UNDERSTANDING MODERN TECHNIQUES IN OPTIMIZATION: FRANK-WOLFE, NESTEROV'S MOMENTUM, AND POLYAK'S MOMENTUM

A Dissertation
Presented to
The Academic Faculty

By

Jun-Kun Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science
College of Computing

Georgia Institute of Technology

May 2021

UNDERSTANDING MODERN TECHNIQUES IN OPTIMIZATION: FRANK-WOLFE, NESTEROV'S MOMENTUM, AND POLYAK'S MOMENTUM

Thesis committee:

Dr. Jacob Abernethy
School of Computer Science
Georgia Institute of Technology

Dr. Richard Peng
School of Computer Science
Georgia Institute of Technology

Dr. Guanghui Lan
Industrial & System Engineering
Georgia Institute of Technology

Dr. Santosh Vempala
School of Computer Science
Georgia Institute of Technology

Dr. Vidya Muthukumar
School of Electrical and Computer Engineering
Industrial & System Engineering
Georgia Institute of Technology

Date approved: March 23, 2021

To my parents

ACKNOWLEDGMENTS

I must begin by expressing my deep gratitude to my advisor, Dr. Jacob Abernethy. Jake has been an outstanding mentor, and he always cares about his students. He gave me a lot of constructive feedback in research and life, and he is very supportive. His patience and advice helped me greatly improve my writing of research papers and improve the skills of giving presentations. The five years of my Ph.D. journey were not possible without him. I will always remember the discussion during which Jake came up with the idea of interpreting Frank-Wolfe as playing a two-player game; I was fortunate in my first year of Ph.D. to learn how Jake identified a connection, wrote down the idea on a note, and managed to meet a conference deadline. I feel this experience helps me a lot in doing research, especially for developing the skills of seeing connections between different algorithms and connections between solving different problems. I also thank him gave me a great deal of freedom in my Ph.D. research.

The five years of Ph.D. are some of my happiest and enjoyable experience, and this is largely due to Jake and also the professors and peers I collaborated with. I want to thank my other collaborators: Bhuvesh Kumar, Chi-Heng Lin, Guanghui Lan, Kevin Lai, and Kfir Levy for the valuable discussions. I also want to thank my committee members Dr. Guanghui Lan, Dr. Richard Peng, Dr. Santosh Vempala, and Dr. Vidya Muthukumar for giving advice.

Finally, I cannot thank my family enough for always supporting me and encouraging me. My mom and my dad give me love and caring every step of the way. I am grateful for their love and support. I also thank my sister for her encouragement. I always feel happy and fortunate when my parents, my sister, and I have a chat online over the past five years. I love my family dearly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Optimization is essential in machine learning, statistics, and data science. Among the first-order optimization algorithms, the popular ones include the Frank-Wolfe method, Nesterov's accelerated methods, and Polyak's momentum. While theoretical analysis of the Frank-Wolfe method and Nesterov's methods are available in the literature, the analysis can be quite complicated or less intuitive. Polyak's momentum, on the other hand, is widely used in training neural networks and is currently the default choice of momentum in Pytorch and Tensorflow. It is widely observed that Polyak's momentum helps to train a neural network faster, compared with the case without momentum. However, there are very few examples that exhibit a provable acceleration via Polyak's momentum, compared to vanilla gradient descent. There is an apparent gap between the theory and the practice of Polyak's momentum.

In the first part of this dissertation research, we develop a modular framework that can serve as a recipe for constructing and analyzing iterative algorithms for convex optimization. Specifically, our work casts optimization as iteratively playing a two-player zero-sum game. Many existing optimization algorithms including Frank-Wolfe and Nesterov's acceleration methods can be recovered from the game by pitting two online learners with appropriate strategies against each other. Furthermore, the sum of the weighted average regrets of the players in the game implies the convergence rate. As a result, our approach provides simple alternative proofs to these algorithms. Moreover, we demonstrate that our approach of "optimization as iteratively playing a game" leads to three new fast Frank-Wolfe-like algorithms for some constraint sets, which further shows that our framework is indeed generic, modular, and easy-to-use.

In the second part, we develop a modular analysis of provable acceleration via Polyak's momentum for certain problems, which include solving the classical strongly quadratic convex problems, training a wide ReLU network under the neural tangent kernel regime,

and training a deep linear network with an orthogonal initialization. We develop a meta theorem and show that when applying Polyak's momentum for these problems, the induced dynamics exhibit a form where we can directly apply our meta theorem.

In the last part of the dissertation, we show another advantage of the use of Polyak's momentum — it facilitates fast saddle point escape in smooth non-convex optimization. This result, together with those of the second part, sheds new light on Polyak's momentum in modern non-convex optimization and deep learning.

CHAPTER 1

INTRODUCTION

1.1 Bridging classical convex optimization and online learning via Fenchel game

In machine learning and data science, training a model is essentially solving an optimization problem,

$$\min_{w \in \mathcal{K}} f(w), \tag{1.1}$$

where w is a vector that represents a model, $\mathcal{K} \subseteq \mathbb{R}^d$ is a constraint set, and $f(\cdot)$ is an objective function which is typically a loss function, e.g. prediction errors of the model over a training dataset. In other words, we are searching for the best model w satisfying $\mathcal{K}$ that minimizes the objective value.

For $f(\cdot)$ being convex, there are many well-established results in optimization literature and quite a few textbooks cover the results well, see e.g. Bertsekas, Nedic, and Ozdaglar [27], Ben-Tal and Nemirovski [26], Hiriart-Urruty and Lemarechal [130], Rockafellar [232], Nesterov [207], Boyd and Vandenberghe [30], Borwein and Lewis [29]. Online learning (a.k.a. no-regret learning), on the other hand, is a growing and an active research area in machine learning, see e.g. Littlestone and Warmuth [177], Cesa-Bianchi and Lugosi [48], Shalev-Shwartz [235], Hazan [126], Orabona [216], Rakhlin and Sridharan [225]. The standard protocol in online learning is that in each round t, the learner must select a point $x_t \in \mathcal{K}$, where $\mathcal{K}$ is her decision space. Then the learner is charged a loss $\ell_t(x_t)$ and typically can observe the loss function $\ell_t(\cdot)$ after she takes an action x_t. The objective of interest in most of the online learning literature is the learner's *regret*, defined as

$$\textsc{Reg}_T^x := \sum_{t=1}^{T} \ell_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(x). \tag{1.2}$$

The goal of the learner is to minimize her regret and to compete with the comparator who foresees all the loss functions and commits to a fixed action.

Algorithm 1 Online-to-batch conversion (adapted from the presentation of Luo [189])

1: **Input:** number of rounds T.

2: **Input:** Training data $\{s_1, s_2, \ldots, s_T\}$ and an online learning algorithm OAlg^x.

3: **for** $t = 1, 2, \ldots, T$ **do**

4: Let x_t be the action of the online learning algorithm OAlg^x at t.

5: Feed OAlg^x with $\ell_t(\cdot) := f(\cdot; s_t)$ as the loss function, where $f(\cdot; s_t)$ is a loss function associated with a sample s_t.

6: **end for**

7: **Output:** $\bar{x} = \frac{\sum_{t=1}^{T} x_t}{T}$

A natural question is "Can we apply an online learning algorithm to solving an offline problem (1.1) and also obtain some theoretical guarantees?". The answer is yes and there is a way to achieve this goal by a technique called the "Online-to-Batch Conversion", which is shown on Algorithm 1 and has the following guarantee (see also e.g. Cesa-Bianchi, Conconi, and Gentile [47] and Appendix B in Shalev-Shwartz [235]).

Theorem 1 (Adapted from the presentation of Luo [189])**.** *Assume* $\mathbb{E}_{s \in \mathcal{D}}[f(\cdot; s)]$ *is convex. Then, with probability* $1 - \delta$*, the online-to-batch conversion (Algorithm 1) guarantees that*

$$\mathbb{E}_{s \sim D} f(\bar{x}, s) \leq \mathbb{E}_{s \sim D} f(x_*; s) + \frac{\textsc{Reg}_T^x}{T} + 2\sqrt{\frac{2\log(2/\delta)}{T}},$$

where $x_* \in \arg\min_{x \in \mathcal{K}} \mathbb{E}_{s \sim D} f(x; s)$.

Theorem 1 says that the average regret of the player, $\frac{\textsc{Reg}_T^x}{T}$, gives a bound of the convergence rate for solving the offline problem (1.1).

It is well known in the literature that if the online loss functions $\{\ell_t(\cdot)\}$ are convex, then the optimal regret is $O\left(\sqrt{T}\right)$; if the online loss functions are strongly convex, then

$O(\log(T))$ is achievable, see e.g. Shalev-Shwartz [235], Cesa-Bianchi and Lugosi [48], Rakhlin and Sridharan [225]. By Theorem 1, these regret bounds imply a convergence rate $O\left(\frac{1}{\sqrt{T}}\right)$ or $O\left(\frac{\log(T)}{T}\right)$ for solving (1.1) when we *convert* an online learning algorithm to an offline one. However, in the optimization literature, there are algorithms that achieve an accelerated rate $O\left(\frac{L}{T^2}\right)$ for solving a L-smooth convex optimization problem, e.g. Nesterov's methods ([205, 204, 208, 207]). Furthermore, when an underlying problem is both μ-strongly convex and L-smooth, the optimal convergence rate is $O\left(\exp\left(-\frac{T}{\sqrt{\kappa}}\right)\right)$, where $\kappa := \frac{L}{\mu}$ is the condition number of the underling function $f(\cdot)$, see e.g. Lan [160]. The gap implies that (offline) optimization and online learning have not been well-connected yet. In this thesis, we will show how to bridge optimization and online learning in a modular and unified way.

Our contributions: Our approach of connecting offline convex optimization and online learning is based on iteratively solving the following two-player game which we call the *Fenchel Game*. We define the payoff function of the game $g : \mathcal{K} \times \mathbb{R}^d$ as follows:

$$g(x, y) := \langle x, y \rangle - f^*(y), \tag{1.3}$$

where $f(\cdot)$ is the underlying function of (1.1) and $f^*(\cdot)$ is the conjugate of $f(\cdot)$, defined as $f^*(y) := \sup_{x \in \mathrm{dom}(f)} \langle x, y \rangle - f(x)$. In this game, the y-player tries to maximize the payoff function $g(\cdot, \cdot)$, while the x-player tries to minimize it. The equilibrium of this game is $\min_{w \in \mathcal{K}} f(w)$ under the assumption that $f(\cdot)$ is convex and lower semi-continuous. Therefore, approximately solving the game is equivalent to approximately solving the offline convex problem (1.1).

This game perspective provides a modular framework for designing and analyzing offline convex optimization algorithms. We will show that several algorithms together with their convergence rates can be recovered from our approach of *optimization as iteratively playing a game*. The algorithms that we will recover include Frank-Wolfe [91] and its several variants [169, 159, 203, 186], Nesterov's accelerated methods [205, 204, 208, 207] and their variants, Heavy Ball [223], and the accelerated proximal method [25]. In particular, we show that the tools and techniques in *online learning* can actually be used to design accelerated algorithms in offline convex optimization. We will establish the accelerated rate $O\left(\frac{1}{T^2}\right)$ and the accelerated linear rate $O\left(\exp\left(-\frac{T}{\sqrt{\kappa}}\right)\right)$ by using the regret analysis with an appropriate weighting scheme.

Most importantly, our insight of *optimization as iteratively playing a game* leads to three new fast Frank-Wolfe-like algorithms for certain constraints sets. Specifically, we propose a Frank-Wolfe-like algorithm that works for non-smooth convex functions *without* using the techniques of smoothing [78] (Algorithm 12), an accelerated $O\left(\frac{1}{T^2}\right)$ rate Frank-Wolfe-like algorithm for smooth convex problems with constraint sets satisfying a notion called strongly convex (Algorithm 14), and a fast parallelizable projection-free algorithm for the nuclear-norm-ball constraint (Algorithm 15). The introduction of the new algorithms verifies that our approach is indeed very modular.

Our results are summarized in Table 2.1 and Table 2.2 in Chapter 2. The materials of Chapter 2 are based on the following papers.

- "On Frank-Wolfe and Equilibrium Computation".
  Jacob Abernethy and Jun-Kun Wang. NeurIPS 2017 (Spotlight).

- "Faster Rates for Convex-Concave Games".
  Jacob Abernethy, Kevin Lai, Kfir Levy, and Jun-Kun Wang. COLT 2018.

- "Acceleration through Optimistic No-Regret Dynamics".
  Jun-Kun Wang and Jacob Abernethy. NeurIPS 2018 (Spotlight).

- "A Fast Parallelizable Projection-Free Algorithm for the Nuclear-Norm-Ball Constraint". Jun-Kun Wang, Bhuvesh Kumar, Jacob Abernethy, and Guanghui Lan.

1.2 Acceleration via Polyak's momentum in deep learning

Polyak's momentum (Algorithm 17 and Algorithm 18) is very popular nowadays for training neural networks and it is the default choice of momentum in PyTorch and Tensorflow. The success of Polyak's momentum in deep learning is widely appreciated and almost all of the recently-developed adaptive gradient methods like Adam [151] and AMSGrad [229] adopt the use of Polyak's momentum, in favor of Nesterov's momentum.

Despite its empirical success in modern machine learning, there is limited theory showing any advantage over vanilla gradient descent. As far as we know, the strongly quadratic convex problem is perhaps the only known example such that *discrete-time* Polyak's momentum has a provable acceleration in terms of the *global convergence* compared with vanilla gradient descent. Most of the existing results (e.g. [223, 168]) only establish a convergence rate in the limit, which is due to the use of Gelfand's formula [107] for approximating the spectral norm of a matrix by its spectral radius. In other words, these results fail to explain the behavior of Polyak's momentum in the non-asymptotic regime even for the classical strongly quadratic convex problems. Moreover, before our work, we are not aware of any theoretical works showing any provable acceleration of Polyak's momentum over vanilla GD in deep learning. Understanding Polyak's momentum remains elusive even though empirically Polyak's momentum appears to provide acceleration in modern machine learning problems.

Our contributions: In Chapter 3, we will develop a modular analysis of Polyak's momentum when applied to the following problems.

- **Strongly convex quadratic problems**

  The objective is

$$\min_{w\in\mathbb{R}^d} \tfrac{1}{2} w^\top \Gamma w + b^\top w, \tag{1.4}$$

  where $\Gamma \in \mathbb{R}^{d\times d}$ is a symmetric matrix such that $\lambda_{\min}(\Gamma) > 0$. We can define the condition number as

$$\kappa^{\mathrm{SC}} := \frac{\lambda_{\max}(\Gamma)}{\lambda_{\min}(\Gamma)}.$$

- (**Training a wide ReLU network with the squared loss**)

  We will consider training the following ReLU network by Polyak's momentum,

$$\mathcal{N}_W^{\mathrm{ReLU}}(x) := \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\langle w^{(r)}, x \rangle), \tag{1.5}$$

  where $\sigma(z) := z \cdot \mathbb{1}\{z \geq 0\}$ is the ReLU activation, $w^{(1)}, \ldots, w^{(m)} \in \mathbb{R}^d$ are the weights of m neurons on the first layer, $a_1, \ldots, a_m \in \mathbb{R}$ are weights on the second layer, and $\mathcal{N}_W^{\mathrm{ReLU}}(x) \in \mathbb{R}$ is the output predicted on input x.

  Giving n number of training samples, following [76, 17, 244], we define a Gram matrix $H \in \mathbb{R}^{n \times n}$ for the weights W, and its expectation $\bar{H} \in \mathbb{R}^{n \times n}$ over the random draws of $w^{(r)} \sim N(0, I_d) \in \mathbb{R}^d$, as follows,

$$\begin{aligned} H(W)_{i,j} &= \sum_{r=1}^{m} \frac{x_i^\top x_j}{m} \mathbb{1}\{\langle w^{(r)}, x_i \rangle \geq 0 \ \& \ \langle w^{(r)}, x_j \rangle \geq 0\} \\ \bar{H}_{i,j} &:= \mathop{\mathbb{E}}_{w^{(r)}} [x_i^\top x_j \mathbb{1}\{\langle w^{(r)}, x_i \rangle \geq 0 \ \& \ \langle w^{(r)}, x_j \rangle \geq 0\}]. \end{aligned} \tag{1.6}$$

  The matrix $\bar{H}$ is also called a neural tangent kernel (NTK) matrix in the literature (e.g. [138, 283, 28]). We can denote the condition number of the neural tangent kernel matrix $\bar{H}$ as

$$\kappa^{\mathrm{ReLU}} := \frac{\lambda_{\max}(\bar{H})}{\lambda_{\min}(\bar{H})}.$$

- (**Training a deep linear network with the squared loss**)

  We will also consider training the following deep linear network by Polyak's momentum,

$$\mathcal{N}_W^{L\text{-linear}}(x) := \frac{1}{\sqrt{m^{L-1} d_y}} W^{(L)} W^{(L-1)} \cdots W^{(1)} x, \tag{1.7}$$

where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ is the weight matrix of the layer $l \in [L]$, and $d_0 = d$, $d_L = d_y$ and $d_l = m$ for $l \neq 1, L$. Let

$$H_t := \frac{1}{m^{L-1}d_y} \sum_{l=1}^{L} [(W_t^{(l-1:1)}X)^\top (W_t^{(l-1:1)}X) \otimes W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top] \in \mathbb{R}^{d_y n \times d_y n}.$$

We will denote the condition number of H_0 as

$$\kappa^{\text{L-linear}} := \frac{\lambda_{\max}(H_0)}{\lambda_{\min}(H_0)}.$$

Theorem 2. *(Informal; see Chapter 3) By setting the momentum parameter β and η appropriately, Polyak's momentum (Algorithm 17 and Algorithm 18) for the three problems aforementioned has*

$$\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \left(1 - \frac{1}{4\sqrt{\kappa}}\right)^t \cdot 8\sqrt{\kappa} \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|,$$

where ξ_t is some residual vector and $\kappa = \{\kappa^{SC}, \kappa^{ReLU}, \kappa^{L\text{-}linear}\}$ is the condition number of the underlying problem.

Our theorem shows the advantage of Polyak's momentum over vanilla gradient descent, as the convergence rate depends on the square root of the condition number $\sqrt{\kappa}$, while the rate of vanilla GD has a dependency on κ. Our work hence shows that Polyak's momentum does improve the neural net training at least for the two canonical models.

Chapter 3 of this thesis is based on the following paper.

- "A Modular Analysis of Provable Acceleration via Polyak's momentum: Training a Wide ReLU Network and a Deep Linear Network" Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. arXiv:2010.01618. 2021

1.3 Exploiting negative curvatures via stochastic Polyak's momentum:

In smooth non-convex optimization, when the iterate enters a region of strict saddle points, defined as

$$\left\{w \in \mathrm{dom}(f) : \|\nabla f(w)\| \leq \epsilon \text{ and } \nabla^2 f(w) \preceq -\epsilon I\right\}, \tag{1.8}$$

the optimization progress slows down. Therefore, it is very important to quickly *escape* the saddle point region. In the literature, there are specialized algorithms designed to exploit the negative curvature explicitly and can escape the saddle point region faster than alternative methods (e.g. [46, 6, 9, 282]). There are also simple GD/SGD variants with minimal tweaks of standard GD/SGD (e.g. [105, 171, 86, 145, 147, 146, 63, 246]). However, none of these works study SGD with Polyak's momentum for escaping saddle points.

Our contributions: We will show that, under certain assumption and some minor constraints that upper-bound parameter β, if SGD with Polyak's momentum has some properties, then we demonstrates that a larger momentum parameter β can help in escaping saddle points faster. Some experiments are provided to support our theoretical results. As saddle points are pervasive in the loss landscape of optimization in deep learning ([68, 55]), this result could help to explain why SGD with momentum enables training faster in optimization for deep learning. We then provide some empirical findings showing that over-parametrization, which is another popular technique in modern machine learning, can help gradient descent exploit negative curvature in the so-called phase retrieval problem. Some discussions are provided in the end.

Chapter 4 of this thesis is based on the following paper.

- "Escaping Saddle Points Faster with Stochastic Momentum" Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. ICLR. 2020.

CHAPTER 2

FENCHEL GAME: A MODULAR APPROACH OF SOLVING CONVEX OPTIMIZATION VIA ITERATIVELY PLAYING A TWO-PLAYER GAME

2.1 Introduction

The main goal of this work is to develop a framework for solving convex optimization problems using iterative methods. Given a convex function $f : \mathbb{R}^d \to \mathbb{R}$, domain $K \subset \mathbb{R}^d$, and some tolerance $\epsilon > 0$, we want to find an approximate minimizer $x \in \mathcal{K}$ so that $f(x) - \min_{x' \in \mathcal{K}} f(x') \leq \epsilon$, using a sequence of oracle calls to f and its derivatives. This foundational problem has received attention for decades, and researchers have designed numerous methods for this problem under a range of oracle query models and structural assumptions on $f(\cdot)$. What we aim to show in this chapter is that a surprisingly large number of these methods—including those of Nesterov [205, 204, 208, 206, 207], Frank and Wolfe [91], Polyak [223], and Beck and Teboulle [25]—can all be described and analyzed through a single unified algorithmic framework, which we call the *Fenchel game no-regret dynamics* (FGNRD). We show that several novel methods, with fast rates, emerge from FGNRD as well.

Let us give a short overview before laying out the FGNRD framework more precisely. A family of tools, largely developed by researchers in theoretical machine learning, consider the problem of sequential prediction and decision making in non-stochastic environments, often called *adversarial online learning*. This online learning setting has found numerous applications in several fields beyond machine learning—finance, for example, as well as statistics—but it has also emerged as a surprisingly useful tool in game theory. What we call *no-regret online learning algorithms* are particularly well-suited for computing equilibria in two-player zero-sum games, as well as solving saddle point problems more broadly. If each agent employs a no-regret online learning algorithm to choose their action at each of a sequence of rounds, it can be shown that the agents' choices will converge to a saddle point, and at a rate that depends on their choice of learning algorithm. Thus, if we are able to simulate the two agents' sequential strategies, where each aims to minimize the "regret" of their chosen actions, then what emerges from the resulting *no-regret dynamics* (NRD) can be implemented explicitly as an algorithm for solving min-max problems.

How does NRD help us to develop and analyze methods for minimizing a convex f? What is our main focus in the present work is a particular game of interest which we call the *Fenchel game*: from f we can construct a two-input "payoff" function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined by

$$g(x, y) := \langle x, y \rangle - f^*(y).$$

We view this as a game in the sense that if one player selects an action x and a second player selects action y, then $g(x, y)$ is the former's "cost" and the latter's "gain" associated to their decisions. If the two players continue to update their decisions sequentially, first choosing x_1 and y_1 then x_2 and y_2, etc., and each player relies on some no-regret algorithm for this purpose, then one can show that the time-averaged iterates $\bar{x}, \bar{y}$ form an approximate equilibrium of the Fenchel game—that is, $g(\bar{x}, y') - \epsilon \leq g(\bar{x}, \bar{y}) \leq g(x', \bar{y}) + \epsilon$ for any alternative x', y'. But indeed, this approximate equilibrium brings us right back to where we started, since using the construction of the Fenchel game it is easy to show that $\bar{x}$ then satisfies $f(\bar{x}) - \min_{x \in \mathcal{K}} f(x) \leq \epsilon$. The approximation factor ϵ is important, and we will see that it depends upon the number of iterations of the dynamic and the players' strategies.

What FGNRD gives us is a recipe book for constructing and analyzing iterative algorithms for convex optimization. To simulate a dynamic we still need to make particular choices as for both players' strategies and analyze their performance. We begin in Section 2.3 by giving a brief overview of tools from adversarial online learning, and we introduce a handful of simple online learning algorithms, including variants of FollowTheLeader and OnlineMirrorDescent, and prove bounds on the *weighted* regret—we generalize slightly the notion of regret by introducing weights $\alpha_t > 0$ for each round. We will also prove a key result that relates the error ϵ of the approximate equilibrium pair $\bar{x}, \bar{y}$, which are the weighted-average of the iterates of the two players, to the weighted regret of the players' strategies. In Section 2.5 we show how several algorithms, including the Heavy Ball method [223], Frank-Wolfe's method [91], and several variants of Nesterov Accelerated Gradient Descent [205, 204, 208, 206, 207, 25], are all special cases of the FGNRD framework, all with special choices of the learning algorithms for the $x-$ and $y-$players, and the weights α_t; see Table 2.1 for a summary of these recipes. In addition we provide several new algorithms using FGNRD in Section 2.6, summarized in Table 2.2.

2.2 Preliminaries

We summarize some results in convex analysis that will be used in this chapter. We also refer the readers to some excellent textbooks (e.g. [27, 130, 232, 207, 30, 29]).

Smoothness and strong convexity A function $f(\cdot)$ on $\mathbb{R}^d$ is L-smooth with respect to a norm $\|\cdot\|$ if $f(\cdot)$ is everywhere differentiable and it has Lipschitz continuous gradient $\|\nabla f(x) - \nabla f(z)\|_* \leq L\|x - z\|$, where $\|\cdot\|_*$ denotes the dual norm. A function $f(\cdot)$ is μ-strongly convex w.r.t. a norm $\|\cdot\|$ if the domain of $f(\cdot)$ is convex and that $f(\theta x + (1-\theta)z) \leq \theta f(x) + (1-\theta)f(z) - \frac{\mu}{2}\theta(1-\theta)\|x - z\|^2$ for all $x, z \in \text{dom}(f)$ and $\theta \in [0, 1]$. If a function is μ-strongly convex, then $f(z) \geq f(x) + \partial f(x)^\top (z - x) + \frac{\mu}{2}\|z - x\|^2$ for all $x, z \in \text{dom}(f)$, where $\partial f(x)$ denotes a subgradient of f at x.

Convex function and conjugate For any convex function $f(\cdot)$, its Fenchel conjugate is

$$f^*(y) := \sup_{x \in \text{dom}(f)} \langle x, y \rangle - f(x) \tag{2.1}$$

If a function $f(\cdot)$ is convex, then its conjugate $f^*(\cdot)$ is also convex, as it is a supremum over linear functions. Furthermore, if the function $f(\cdot)$ is closed and convex, the following are equivalent: (I) $y \in \partial f(x)$, (II) $x \in \partial f^*(y)$, and (III)

$$\langle x, y \rangle = f(x) + f^*(y), \tag{2.2}$$

which also implies that the biconjudate is equal to the original function, i.e. $f^{**}(\cdot) = f(\cdot)$. Moreover, when the function $f(\cdot)$ is differentiable, we have $\nabla f(x) = \sup_y \langle x, y \rangle - f^*(y)$. We refer to the readers to Bauschke and Lucet [23], Kakade, Shalev-shwartz, and Tewari [148], and textbooks (e.g. [232, 30, 29]) for more details of Fenchel conjugate. Througout this chapter, unless specifically mentioned, we assume that the underlying convex function is proper, closed, and differentiable.

An important property of a closed and convex function is that $f(\cdot)$ is L-smoooth w.r.t. some norm $\|\cdot\|$ if and only if its conjugate $f^*(\cdot)$ is $1/L$-strongly convex w.r.t. the dual norm $\|\cdot\|_*$ (e.g. Theorem 6 in Kakade, Shalev-shwartz, and Tewari [148]).

Bregman Divergence. We will denote the Bregman divergence $D_z^\phi(\cdot)$ centered at a point z with respect to a β-strongly convex distance generating function $\phi(\cdot)$ as

$$D_z^\phi(x) := \phi(x) - \langle \nabla\phi(z), x - z \rangle - \phi(z). \tag{2.3}$$

Strongly convex sets. A convex set $\mathcal{K} \subseteq \mathbb{R}^m$ is an λ-*strongly convex set* w.r.t. a norm $\|\cdot\|$ if for any $x, z \in \mathcal{K}$, any $\theta \in [0, 1]$, the $\|\cdot\|$ ball centered at $\theta x + (1-\theta)z$ with radius $\theta(1-\theta)\frac{\lambda}{2}\|x - z\|^2$ is included in $\mathcal{K}$ [100]. Examples of strongly convex sets include ℓ_p balls: $\|x\|_p \leq r, \forall p \in (1, 2]$, Schatten p balls: $\|\sigma(X)\|_p \leq r$ for $p \in (1, 2]$, and Group (s,p) balls: $\|X\|_{s,p} = \|(\|X_1\|_s, \|X_2\|_s, \ldots, \|X_m\|_s)\|_p \leq r$ (see e.g. [100]).

Min-max problems and (approximate) Nash equilibrium A large number of core problems in statistics, optimization, and machine learning, can be framed as the solution of a two-player zero-sum game. Linear programs, for example, can be viewed as a competi-

Table 2.1: Summary of recovering existing optimization algorithms from *Fenchel Game*. Here T denotes the total number of iterations, α_t are the weights which set the emphasis on iteration t, the last two columns on the table indicate the specific strategies of the players in the FGNRD.

L-Smooth convex optimization: $\min_w f(w)$ as a game $g(x,y) := \langle x, y\rangle - f^*(y)$.

Algorithm	rate	weight	y-player	x-player
Frank-Wolfe method [91]	Thm. 5 and 6 $O(\frac{L\log T}{T})$	$\alpha_t = 1$	Sec. 2.4.1 FTL	Sec. 2.4.8 BESTRESP$^+$
Frank-Wolfe method [91]	Thm. 5 and 6 $O(\frac{L}{T})$	$\alpha_t = t$	Sec. 2.4.1 FTL	Sec. 2.4.8 BESTRESP$^+$
Linear rate FW [169]	Thm. 7 $O(\exp(-\frac{\lambda T}{L}))$	$\alpha_t = \frac{1}{\Vert\ell_t(x_t)\Vert^2}$	Sec. 2.4.1 FTL	Sec. 2.4.8 BESTRESP$^+$
Nesterov's (1- memory) method [206]	Thm. 9 and 10 $O(\frac{L}{T^2})$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.9 OMD$^+$
Nesterov's (∞- memory) method [208]	Thm. 9 and 10 $O(\frac{L}{T^2})$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.5 FTRL$^+$
Nesterov's first acceleration method [204]	Thm. 9 and 11 $O(\frac{L}{T^2})$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.9 OMD$^+$ with $\phi_t(x) = \frac{1}{2}\Vert x\Vert_2^2$
Heavy Ball method [223]	Thm. 12 $O(\frac{L}{T})$	$\alpha_t = t$	Sec. 2.4.1 FTL	Sec. 2.4.5 FTRL$^+$

Non-smooth convex optimization: $\min_w f(w)$ as a game $g(x,y) := \langle x, y\rangle - f^*(y)$.

Algorithm	rate	weight	y-player	x-player
Smoothed FW [159]	$O(\frac{1}{\sqrt{T}})$	$\alpha_t = 1$	Sec. 2.4.7 FTPL	Sec. 2.4.8 BESTRESP$^+$

Composite optimization: $\min_w f(w) + \psi(w)$, where $\psi(\cdot)$ is possibly non-differentiable, as a game $g(x,y) := \langle x, y\rangle - f^*(y) + \psi(x)$.

Algorithm	rate	weight	y-player	x-player
Accelerated proximal method [25]	Thm. 9 and 13 $O(\frac{L}{T^2})$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.9 OMD$^+$

L-smooth and μ strongly convex optimization: $\min_w f(w)$ as a game $g(x,y) := \langle x, y\rangle - \tilde{f}^*(y) + \frac{\mu\Vert x\Vert^2}{2}$, where $\tilde{f}(\cdot) := f(\cdot) - \frac{\mu}{2}\Vert\cdot\Vert^2$.

Algorithm	rate	α_t	y-player	x-player
Nesterov's method [207]	Thm. 14 $O(\exp(-\sqrt{\frac{\mu}{L}}T))$	$\alpha_t \propto \exp(t)$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.5 FTRL$^+$

tion between a feasibility player, who selects a point in $\mathbb{R}^n$, and a constraint player that aims to check for feasibility violations [5]. Boosting [95] can be viewed as the competition between an agent that selects hard distributions and a weak learning oracle that aims to

Table 2.2: Summary of *new* optimization algorithms from *Fenchel Game*. Here T denotes the total number of iterations, α_t are the weights which set the emphasis on iteration t, the last two columns on the table indicate the specific strategies of the players in the FGNRD.

Non-smooth convex optimization: $\min_{w \in \mathcal{K}} f(w)$, where $\mathcal{K}$ is a λ-strongly convex set as a game $g(x,y) := \langle x, y \rangle - f^*(y)$.
Assume that the norm of cumulative gradient does not vanish, $\| \frac{1}{t} \sum_{s=1}^{t} \partial f(x_s) \| \geq \rho$.

Algorithm	rate	weight	y-player's alg.	x-player's alg.
Boundary FW	Thm. 15 $O(\frac{1}{\lambda \rho T})$	$\alpha_t = 1$	Sec. 2.4.1 FTL	Sec. 2.4.8 BESTRESP$^+$

L-smooth convex optimization: $\min_{w \in \mathcal{K}} f(w)$,
where $\mathcal{K}$ is a λ-strongly convex set that is centrally symmetric and contains the origin,
as a game $g(x,y) := \langle x, y \rangle - f^*(y)$

Algorithm	rate	weight	y-player's alg.	x-player's alg.
Gauge FW	Thm. 16 $O(\frac{L}{\lambda T^2})$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	Sec. 2.4.5 FTRL$^+$ with gauge function

L-smooth convex optimization: $\min_{w \in \mathcal{K}} f(w)$,
where $\mathcal{K}$ is a nuclear-norm ball $\left\{ W \in \mathbb{R}^{d_1 \times d_2} : \sum_{i=1}^{d_1 \wedge d_2} \sigma_i(W) \leq r \right\}$
with the spectral norm of the gradient satisfying $\|\nabla f(\cdot)\|_2 \leq G$ for all $W \in \mathcal{NB}_{d_1,d_2}(r)$,
as a game $g(x,y) := \langle x, y \rangle - f^*(y)$

Algorithm	rate	weight	y-player's alg.	x-player's alg.
Parallelizable Projection-Free Alg.	Cor. 2 $\tilde{O}\left(\frac{Lr \log(d_1+d_2)}{T^2} + \frac{G}{T}\right)$	$\alpha_t = t$	Sec. 2.4.3 OPTIMISTICFTL	FTRL$^+$ with a random projection

overcome such challenges [94]. The hugely popular technique of Generative Adversarial Networks (GANs) [119], which produce implicit generative models from unlabelled data, has been framed in terms of a repeated game, with a distribution player aiming to produce realistic samples and a discriminative player that seeks to distinguish real from fake.

Given a zero-sum game with *payoff function* $g(x,y)$ which is convex in x and concave in y, define $V^* = \inf_{x \in \mathcal{K}} \sup_y g(x,y)$. An ϵ-*equilibrium* of $g(\cdot,\cdot)$ is a pair $\hat{x}, \hat{y}$ such that

$$V^* - \epsilon \leq \inf_{x \in \mathcal{K}} g(x, \hat{y}) \leq V^* \leq \sup_y g(\hat{x}, y) \leq V^* + \epsilon. \tag{2.4}$$

The Fenchel Game. One of the core tools of this work is as follows. In order to solve the problem

$$\min_{x \in \mathcal{K}} f(x) \tag{2.5}$$

we instead construct a saddle-point problem which we call the *Fenchel Game*. We define $g : \mathcal{K} \times \mathbb{R}^d$ as follows:

$$g(x, y) := \langle x, y \rangle - f^*(y). \tag{2.6}$$

This payoff function is useful for solving the original optimization problem, since an equilibrium of this game provides us with a solution to $\min_{x \in \mathcal{K}} f(x)$. Let $\hat{x}, \hat{y}$ be any equilibrium pair of g, with $\hat{x} \in \mathcal{K}$. that is, where $V^* = \sup_y g(\hat{x}, y)$. Then we have

$$\begin{aligned} \inf_{x \in \mathcal{K}} f(x) &= \inf_{x \in \mathcal{K}} \sup_y \{\langle x, y \rangle - f^*(y)\} = \inf_{x \in \mathcal{K}} \sup_y g(x, y) \\ &= \sup_y g(\hat{x}, y) = \sup_y \{\langle \hat{x}, y \rangle - f^*(y)\} = f(\hat{x}) \end{aligned}$$

In other words, given an equilibrium pair $\hat{x}, \hat{y}$ of $g(\cdot, \cdot)$, we immediately have a minimizer of $f(\cdot)$. This simple observation can be extended to approximate equilibria as well.

Lemma 1. *If $(\hat{x}, \hat{y})$ is an ϵ-equilibrium of the Fenchel Game* (2.6)*, then $f(\hat{x}) - \min_x f(x) \leq \epsilon$.*

Lemma 1 sets us up for the remainder of the chapter. The framework, which we lay out precisely in Section 2.3.2, will consider two players sequentially playing the Fenchel game, where the y-player sequentially outputs iterates $y_1, y_2, \ldots$, while alongside the x-player returns iterates $x_1, x_2, \ldots$. Each player may use the previous sequence of actions of their opponent in order to choose their next point x_t or y_t, and we will rely heavily on the use of no-regret online learning algorithms described in Section 2.3. In addition, we need to select a sequence of weights $\alpha_1, \alpha_2, \ldots > 0$ which determine the "strength" of each round, and can affect the players' update rules. What we will be able to show is that the α-weighted average iterate pair, defined as

$$(\hat{x}, \hat{y}) := \left(\frac{\alpha_1 x_1 + \ldots + \alpha_T x_T}{\alpha_1 + \cdots + \alpha_T}, \frac{\alpha_1 y_1 + \cdots + \alpha_T y_T}{\alpha_1 + \cdots + \alpha_T} \right),$$

is indeed an ϵ-equilibrium of $g(\cdot, \cdot)$, and thus via Lemma 1 we have that $\hat{x}$ approximately minimizes f. To get a precise estimate of ϵ requires us to prove a family of regret bounds, which is the focus of the following section.

2.3 No-regret learning algorithms

An algorithmic framework, often referred to as *no-regret learning* or *online convex optimization*, has been developed mostly within the machine learning research community, has grown quite popular as it can be used in a broad class of sequential decision problems. As we will explain in Section 2.3.1, one imagines an algorithm making repeated decisions by selecting a vector of parameters in a convex set, and on each round is charged according to a varying convex loss function. The algorithm's goal is to minimize an objective known as regret. In Section 2.3.2, we describe how online convex optimization algorithms with vanishing regret can be implemented in a two-player protocol which sequentially computes an approximate equilibria for a convex-concave payoff function. This is the core tool that allows us to describe a range of known and novel algorithms for convex optimization, by modularly combining pairs of OCO strategies. In Section 2.4 we provide several such OCO algorithms, most of which have been proposed and analyzed over the past 10-20 years.

Protocol 2 Weighted Online Convex Optimization

```
Input: decision set K ⊂ R^n
Input: number of rounds T
Input: weights α_1, α_2, ..., α_T > 0          # Weights determined in advance
Input: algorithm OAlg                          #This implements the learner's update strategy
for t = 1, 2, ..., do
    Return: x_t ← OAlg                          #Alg returns a point x_t
    Receive: α_t, ℓ_t(·) → OAlg                 #Alg receives loss fn. and round weight
    Evaluate: Loss ← Loss + α_t ℓ_t(x_t)        #Alg suffers weighted loss for choice of x_t
end for
```

2.3.1 Online Convex Optimization and Regret

Here we describe the framework, given precisely in Protocol 2, for online convex optimization. We assume we have some learning algorithm known as OAlg that is tasked with selecting "actions" from a compact and convex *decision set* $\mathcal{K} \subset \mathbb{R}^d$. On each round $t = 1, \ldots, T$, OAlg returns a point $x_t \in \mathcal{K}$, and is then presented with the pair α_t, ℓ_t, where $\alpha_t > 0$ is a weight for the current round and $\ell_t : \mathcal{K} \to \mathbb{R}$ is a convex loss function that evaluates the choice x_t. While OAlg is essentially forced to "pay" the cost $\alpha_t \ell_t(x_t)$, it can then update its state to provide better choices in future rounds.

On each round t, the learner must select a point $x_t \in \mathcal{K}$, and is then "charged" a loss of $\alpha_t \ell_t(x_t)$ for this choice. Typically it is assumed that, when the learner selects x_t on round t, she has observed all loss functions $\alpha_1 \ell_1(\cdot), \ldots, \alpha_{t-1}\ell_{t-1}(\cdot)$ up to, but not including, time t. However, we will also consider learners that are *prescient*, i.e. that can choose x_t with knowledge of the loss functions up to *and including* time t. The objective of interest in most of the online learning literature is the learner's *regret*, defined as

$$\boldsymbol{\alpha}\text{-}\textsc{Reg}^x := \sum_{t=1}^{T} \alpha_t \ell_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \alpha_t \ell_t(x). \tag{2.7}$$

Oftentimes we will want to refer to the *average regret*, or the regret normalized by the time weight $A_T := \sum_{t=1}^{T} \alpha_t$, which we will denote $\overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x := \frac{\boldsymbol{\alpha}\text{-}\textsc{Reg}^x}{A_T}$. Note that in online learning literature, what has become a cornerstone of online learning research has been the existence of *no-regret algorithms*, i.e. learning strategies that guarantee $\overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x \to 0$ as $A_T \to \infty$.

Let us consider some very simple learning strategies that will be used in this chapter, and we note the available guarantees for each. We also refer the readers to some tutorial of online learning for more online learning algorithms (see e.g. [216, 225, 127, 236]).

2.3.2 Framework: optimization as *Fenchel Game*

We consider Fenchel game (2.6) with weighted losses depicted in Algorithm 3. In this game, the y-player plays before the x-player plays and the x-player sees what the y-player plays before choosing its action. The y-player receives loss functions $\alpha_t \ell_t(\cdot)$ in round t, in which $\ell_t(y) := f^*(y) - \langle x_t, y \rangle$, while the x-player see its loss functions $\alpha_t h_t(\cdot)$ in round t, in which $h_t(x) := \langle x, y_t \rangle - f^*(y_t)$. Consequently, we can define the *weighted regret* of the x and y players as

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^y := \sum_{t=1}^{T} \alpha_t \ell_t(y_t) - \min_y \sum_{t=1}^{T} \alpha_t \ell_t(y) \tag{2.8}$$

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^x := \sum_{t=1}^{T} \alpha_t h_t(x_t) - \sum_{t=1}^{T} \alpha_t h_t(x^*) \tag{2.9}$$

Notice that the x-player's regret is computed relative to x^* the minimizer of $f(\cdot)$, rather than the minimizer of $\sum_{t=1}^{T} \alpha_t h_t(\cdot)$.

Protocol 3 Fenchel Game No-Regret Dynamics

Input: number of rounds T
Input: decision sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^d$
Input: Convex-concave payoff function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$
Input: weights $\alpha_1, \alpha_2, \ldots, \alpha_T > 0$ Weights determined in advance
Input: algorithms OAlg^Y, OAlg^X #Learning algorithms for both players
for $t = 1, 2, \ldots, T$ **do**
  Return: $y_t \leftarrow$ OAlg^Y #y-player returns a point y_t
  Update: $\alpha_t, h_t(\cdot) \to$ OAlg^X #x-player updates with α_t and loss $-g(\cdot, y_t)$
    where $h_t(\cdot) := -g(\cdot, y_t)$
  Return: $x_t \leftarrow$ OAlg^X #x-player returns a point x_t
  Update: $\alpha_t, \ell_t(\cdot) \to$ OAlg^Y #y-player updates with α_t and loss $g(x_t, \cdot)$
    where $\ell_t(\cdot) := g(x_t, \cdot)$
end for
Output $(\bar{x}_T, \bar{y}_T) := \left(\frac{\sum_{s=1}^{T} \alpha_s x_s}{A_T}, \frac{\sum_{s=1}^{T} \alpha_s y_s}{A_T} \right)$.

At times when we want to refer to the regret on another sequence $y'_1, \ldots, y'_T$ we may refer to this as $\boldsymbol{\alpha}\text{-}\mathrm{REG}(y'_1, \ldots, y'_T)$. We also denote A_t as the cumulative sum of the weights $A_t := \sum_{s=1}^{t} \alpha_s$ and the weighted average regret $\overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}} := \frac{\boldsymbol{\alpha}\text{-}\mathrm{REG}}{A_T}$. Finally, for offline constrained optimization (i.e. $\min_{x\in\mathcal{K}} f(x)$), we let the decision space of the benchmark/-comparator in the weighted regret definition to be $\mathcal{X} = \mathcal{K}$; for offline unconstrained optimization, we let the decision space of the benchmark/comparator to be a norm ball that contains the optimum solution of the offline problem (i.e. contains $\arg\min_{x\in\mathbb{R}^n} f(x)$), which means that $\mathcal{X}$ of the comparator is a norm ball. We let $\mathcal{Y} = \mathbb{R}^d$ be unconstrained.

Theorem 3. *Assume a T-length sequence $\boldsymbol{\alpha}$ are given. Suppose in Algorithm 3 the online learning algorithms OAlg^x and OAlg^y have the $\boldsymbol{\alpha}$-weighted average regret $\overline{\boldsymbol{\alpha}\text{-REG}}^x$ and $\overline{\boldsymbol{\alpha}\text{-REG}}^y$ respectively. Then the output $(\bar{x}_T, \bar{y}_T)$ is an ϵ-equilibrium for $g(\cdot,\cdot)$, with $\epsilon = \overline{\boldsymbol{\alpha}\text{-REG}}^x + \overline{\boldsymbol{\alpha}\text{-REG}}^y$.*

Proof. Suppose that the loss function of the x-player in round t is $\alpha_t h_t(\cdot) : \mathcal{X} \to \mathbb{R}$, where $h_t(\cdot) := g(\cdot, y_t)$. The y-player, on the other hand, observes her own sequence of loss functions $\alpha_t \ell_t(\cdot) : \mathcal{Y} \to \mathbb{R}$, where $\ell_t(\cdot) := -g(x_t, \cdot)$.

$$
\begin{aligned}
\frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T \alpha_t g(x_t, y_t) &= \frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T -\alpha_t \ell_t(y_t) \\
&\geq -\frac{1}{\sum_{s=1}^T \alpha_s} \inf_{y\in\mathcal{Y}} \left\{ \sum_{t=1}^T \alpha_t \ell_t(y) \right\} - \frac{\boldsymbol{\alpha}\text{-REG}^y}{\sum_{s=1}^T \alpha_s} \\
&= \sup_{y\in\mathcal{Y}} \left\{ \frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T \alpha_t g(x_t, y) \right\} - \overline{\boldsymbol{\alpha}\text{-REG}}^y \\
\text{(Jensen)} \quad &\geq \sup_{y\in\mathcal{Y}} g\left(\frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T \alpha_t x_t, y \right) - \overline{\boldsymbol{\alpha}\text{-REG}}^y \qquad (2.10) \\
&= \sup_{y\in\mathcal{Y}} g\left(\bar{x}_T, y\right) - \overline{\boldsymbol{\alpha}\text{-REG}}^y \qquad (2.11) \\
&\geq \inf_{x\in\mathcal{X}} \sup_{y\in\mathcal{Y}} g\left(x, y\right) - \overline{\boldsymbol{\alpha}\text{-REG}}^y
\end{aligned}
$$

Let us now apply the same argument on the right hand side, where we use the x-player's regret guarantee.

$$
\begin{aligned}
\frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T \alpha_t g(x_t, y_t) &= \frac{1}{\sum_{s=1}^T \alpha_s} \sum_{t=1}^T \alpha_t h_t(x_t) \\
&\leq \left\{ \sum_{t=1}^T \frac{1}{\sum_{s=1}^T \alpha_s} \alpha_t h_t(x^*) \right\} + \frac{\boldsymbol{\alpha}\text{-}\textsc{Reg}^x}{\sum_{s=1}^T \alpha_s} \\
&= \left\{ \sum_{t=1}^T \frac{1}{\sum_{s=1}^T \alpha_s} \alpha_t g(x^*, y_t) \right\} + \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x \\
\text{(Jensen)} \quad &\leq g\left(x^*, \sum_{t=1}^T \frac{1}{\sum_{s=1}^T \alpha_s} \alpha_t y_t \right) + \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x \qquad (2.12) \\
&= g\left(x^*, \bar{y}_T\right) + \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x \qquad (2.13) \\
&\leq \sup_{y \in \mathcal{Y}} g(x^*, y) + \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x
\end{aligned}
$$

Note that $\sup_{y\in\mathcal{Y}} g(x^*, y) = f(x^*)$ by Fenchel conjugacy, and hence we can conclude that $\sup_{y\in\mathcal{Y}} g(x^*, y) = V^* = \sup_{y\in\mathcal{Y}} \inf_{x\in\mathcal{X}} g(x, y) = \inf_{x\in\mathcal{X}} \sup_{y\in\mathcal{Y}} g(x, y)$. Combining (2.11) and (2.13), we see that $(\bar{x}_T, \bar{y}_T)$ is an $\epsilon = \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^x + \overline{\boldsymbol{\alpha}\text{-}\textsc{Reg}}^y$ equilibrium. $\square$

In order to utilize minimax duality, we have to define decision sets for two players, and we must produce a convex-concave payoff function. First we will assume, for convenience, that $f(x) := \infty$ for any $x \notin \mathcal{X}$. That is, it takes the value ∞ outside of the convex/compact set $\mathcal{X}$, which ensures that $f(\cdot)$ is lower semi-continuous and convex. Now, let the x-player be given the set $\mathcal{X} := \{\nabla f(x) : x \in \mathcal{X}\}$. One can check that the closure of the set X is a convex set. Section 2.7.1 describes the proof.

Theorem 4. *The closure of (sub-)gradient space $\{\partial f(x) | x \in \mathcal{X}\}$ is a convex set.*

2.4 Online Convex Optimization: An Algorithmic Menu

In this section we introduce and analyze several core online learning algorithms. Later in Sections 2.5 & 2.6, we will show how composing different online learning algorithm within the Fenchel Game No-Regret Dynamics (Protocol 3) enables to easily recover known results and methods for convex optimization (Section 2.5), as well as to design new algorithm with novel guarantees (Section 2.6).

We start by introducing the simplest algorithmic templates, and then move towards more advanced techniques. In Subsection 2.4.6 we introduce and analyze a Meta-algorithm that generalizes many of the methods and results that we introduce in the first subsections. For the sake of generality we provide guarantees assuming that the loss functions are strongly-convex. Setting the strong-convexity parameter to 0 recovers the results for general convex losses.

2.4.1 FTL (Follow The Leader)

FTL (Follow The Leader) is perhaps the simplest strategy in online learning, which plays the best fixed action for the cumulative (weighted) loss seen so far in each round (Equation (2.14)). The corresponding analysis has been shown in many textbooks (e.g. [216, 236]).

Lemma 2. *(**FTL**$[z_{\text{init}}]$) Let $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$ be a sequence of loss functions such that each $\ell_t(\cdot)$ is μ-strongly convex, where $\mu \geq 0$. Given an initial point $z_{\text{init}} \in \mathcal{Z}$, **FTL**$[z_{\text{init}}]$ is defined as follows,*

$$
\begin{aligned}
z_1 &= z_{\text{init}} \\
z_t &= \operatorname*{argmin}_{z \in \mathcal{Z}} \sum_{s=1}^{t-1} \alpha_s \ell_s(z)
\end{aligned}
\tag{2.14}
$$

and satisfies the following regret bound,

$$
\boldsymbol{\alpha}\text{-}\mathrm{REG}^z \leq \textstyle\sum_{t=1}^T \frac{2\alpha_t^2}{\left(\sum_{s=1}^t \alpha_s \mu\right)} \|\nabla \ell_t(z_t)\|_*^2. \tag{2.15}
$$

By Lemma 2, when we set the weights uiniformly, i.e. $\alpha_t = 1\ \forall t$, and assume a bound on the gradient norms, i.e. $\|\nabla \ell_t(z_t)\|_*^2 \leq G$, the uniform regret is

$$\text{REG} := \textstyle\sum_{t=1}^T \ell_t(z_t) - \min_z \sum_{t=1}^T \ell_t(z) \leq \frac{G\log(T+1)}{2\mu}, \tag{2.16}$$

which is a logarithmic regret in T.

On the other hand, when the loss function is linear, i.e. $\ell_t(\cdot) := \langle \theta_t, \cdot \rangle$ for some loss vector $\theta_t \in \mathbb{R}^d$, FTL might suffer linear regret. That is, the uniform regret could be $\text{REG} := \sum_{t=1}^T \ell_t(z_t) - \min_{z\in\mathcal{Z}} \sum_{t=1}^T \ell_t(z) = \Theta(T)$ for some convex polytope $\mathcal{Z}$, which means that the learner fails to learn (see e.g. Example 2.2 [236]). However, if the constraint set $\mathcal{Z}$ satisfies a notion called strongly-convexity, then obtaining a logarithmic regret is possible even when the loss function is linear.

Lemma 3 (Theorem 3.3 in [137])**.** *Let $\{\ell_t(\cdot) := \langle \theta_t, \cdot \rangle\}_{t=1}^T$ be any sequence of linear loss functions. Denote $G := \max_{t\leq T} \|\theta_t\|$ and assume that the support function $\Phi(\cdot) := \max_{z\in\mathcal{Z}}(z, \cdot)$ has a unique maximizer for each cumulative loss vector $L_t := \sum_{s=1}^t \theta_s$ at round t. Define $\nu_T := \min_{1\leq t\leq T} \|L_t\|$. Let $\mathcal{Z} \subset \mathbb{R}^d$ be an λ-strongly convex set. Choose $z_{\text{init}} \in$ boundary$(\mathcal{Z})$. Then, after T rounds,* **FTL** $[\mathbf{z}_{\text{init}}]$ *ensures,*

$$\text{REG} := \sum_{t=1}^T \ell_t(z_t) - \min_{z\in\mathcal{Z}} \sum_{t=1}^T \ell_t(z) = \frac{2G^2}{\lambda \nu_T}(1 + \log(T)). \tag{2.17}$$

2.4.2 FTL$^+$ (Be The Leader)

As can be seen from Equation (2.18), in FTL$^+$ (a.k.a. Be The Leader) the learner plays the best fixed action for the cumulative (weighted) loss seen so far *including the current round*. FTL$^+$ is often used as analytic tool rather than a practical algorithm. Nevertheless, note that in FGNRD (protocol 3) the x-player is allowed to view the current loss prior to playing, and can therefore apply FTL$^+$. This algorithm was named by [149], who also proved that it actually guarantees non-positive regret. Here we provide a tighter bound.

Lemma 4. *(**FTL$^+$**) Let $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$ be a sequence of loss functions such that each $\ell_t(\cdot)$ is at least μ-strongly convex, where $\mu \geq 0$. **FTL$^+$** is defined as follows,*

$$z_t = \text{argmin}_{z \in \mathcal{Z}} \sum_{s=1}^t \alpha_s \ell_s(z). \tag{2.18}$$

and satisfies the following regret bound,

$$\boldsymbol{\alpha}\text{-}\textsc{Reg}^z \leq - \sum_{t=1}^T \frac{\mu A_{t-1}}{2} \|z_{t-1} - z_t\|^2 \leq 0. \tag{2.19}$$

2.4.3 OPTIMISTICFTL

In the previous subsection, we have seen that FTL$^+$ uses the knowledge of the loss function at rounds t in order to ensure negative regret. While this knowledge is oftentimes unavailable, one can often access a "hint" function $m_t(\cdot)$ that approximates $\ell_t(\cdot)$ prior to choosing an action z_t. As can be seen from Equation (2.20) and Lemma 5, OPTIMISTICFTL makes use of the availability of such hints in order to provide better guarantees. The next statement shows that when we have "good" hints, in the sense that $m_t(\cdot) \approx \ell_t(\cdot)$, then OPTIMISTICFTL obtains improved guarantees compared to standard FTL.

Lemma 5. *(**OptimisticFTL**$[z_{\text{init}}]$) Let $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$ be a sequence of loss functions such that each $\ell_t(\cdot)$ is μ_t-strongly convex. Given an initial point $z_{\text{init}} = \arg\min_{z \in \mathcal{Z}} m_1(\cdot)$, **OptimisticFTL**$[z_{\text{init}}]$ is defined as follows,*

$$\begin{aligned} &z_1 \leftarrow z_{\text{init}} \\ &z_t \leftarrow \arg\min_{z \in \mathcal{Z}} \left(\sum_{s=1}^{t-1} \alpha_s \ell_s(z) \right) + \alpha_t m_t(z), \end{aligned} \tag{2.20}$$

where $m_t(\cdot)$ *is the* hint (or the guess) *for the loss function* $\ell_t(\cdot)$.

OPTIMISTICFTL *satisfies,*

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^z \leq \sum_{t=1}^{T} \alpha_t \left(\ell_t(z_t) - \ell_t(w_{t+1})\right) - \alpha_t \left(m_t(z_t) - m_t(w_{t+1})\right) \tag{2.21}$$

where $w_t := \mathrm{argmin}_{z \in \mathcal{Z}} \sum_{s=1}^{t-1} \ell_s(z)$.

2.4.4 FTRL (Follow The Regularized Leader)

FTRL also called dual averaging in optimization literature [280] is a classic algorithm in online learning (see e.g. [216, 127]). Looking at Equation (2.22) one can notice that FTRL is similar to FTL with an additional *Regularization term* $R(\cdot)$ that is scale by a factor $1/\eta$. The regularization term induces *stability* into the decisions of the player, i.e., it enforces consecutive decisions to be close to each other; and this property is often crucial in order to ensure regret guarantees. For example, in the case of linear loss functions, FTRL (with appropriate choices of $\eta, R(\cdot)$) can ensure sublinear regret guarantees, while FTL cannot. In what follows we assume that $R(\cdot)$ is a β-strongly-convex function over $\mathcal{Z}$.

Lemma 6. *(***FTRL**$[R(\cdot), \eta]$*) Let* $\{\alpha_t \ell_t(\cdot)\}_{t=1}^{T}$ *be a sequence of loss functions such that each* $\ell_t(\cdot)$ *is* μ*-strongly convex, where* $\mu \geq 0$*. Also let* $\eta > 0$ *and* $R(\cdot)$ *be a* β*-strongly-convex function over* $\mathcal{Z}$*. Then* **FTRL**$[R(\cdot), \eta]$ *is defined as follows,*

$$z_t = \mathrm{argmin}_{z \in \mathcal{Z}} \textstyle\sum_{s=1}^{t-1} \alpha_s \ell_s(z) + \frac{1}{\eta} R(z) \tag{2.22}$$

and satisfies the following regret bound,

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^z \leq \textstyle\sum_{t=1}^{T} \frac{2\alpha_t^2}{\left(\sum_{s=1}^{t} \alpha_s \mu\right) + \beta} \|\nabla \ell_t(z_t)\|_*^2 + \frac{1}{\eta}\left(R(z^*) - R(z_1)\right). \tag{2.23}$$

2.4.5 FTRL$^+$ (Be The Regularized Leader)

FTRL$^+$ is a very similar to FTRL, with the difference that the former has an access to all past loss functions up to and *including* the current round. Recall that in our FGNRD template (protocol 3) the x-player is allowed to view the current loss prior to playing, and can therefore apply FTRL$^+$.

Lemma 7. *(***FTRL**$^+$ $[R(\cdot), 1/\eta]$*) Let* $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$ *be a sequence of loss functions such that each* $\ell_t(\cdot)$ *is* μ*-strongly convex, where* $\mu \geq 0$*. Also let* $\eta > 0$ *and* $R(\cdot)$ *be a* β*-strongly-convex function over* $\mathcal{Z}$*. Then* **FTRL**$^+[R(\cdot), \eta]$ *is defined as follows,*

$$z_t \leftarrow \arg\min_{z\in\mathcal{Z}} \sum_{s=1}^{t} \alpha_s \ell_s(z) + \frac{1}{\eta} R(z), \tag{2.24}$$

and satisfies the following regret bound,

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^z \leq \tfrac{R(z^*)-R(z_0)}{\eta} - \textstyle\sum_{t=1}^T \left(\frac{\mu A_{t-1}}{2} + \frac{\beta}{2\eta}\right) \|z_{t-1} - z_t\|^2. \tag{2.25}$$

where $z_0 = \min_{z\in\mathcal{Z}} R(z)$ *and* z^* *is any point in* $\mathcal{Z}$*.*

2.4.6 A meta online learning algorithm: OPTIMISTICFTRL

Here we describe OPTIMISTICFTRL, a Meta-algorithm that captures all previously mentioned methods as a private cases. As can be seen from Equation (2.26), OPTIMISTICFTRL employs a regularization term (similarly to FTRL and FTRL$^+$), and makes use of a hint sequence $m_t(\cdot)$ (similarly to OPTIMISTICFTL).

In Lemma 8 we state the regret guarantees of OPTIMISTICFTRL, and then show how does the guarantees of FTL, FTL$^+$, OPTIMISTICFTL, FTRL, and FTRL$^+$ follow as corollaries of this Lemma. The proof of Lemma 8 is provided in Subsection 2.4.10.

Lemma 8. *(***OptimisticFTRL**$[R(\cdot), \eta]$*) Let* $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$ *be a sequence of loss functions such that each* $\ell_t(\cdot)$ *is* μ_t*-strongly convex,* $\mu \geq 0\ \forall t$*. Also let* $\eta > 0$ *and* $R(\cdot)$ *be a* β*-*

strongly-convex function over $\mathcal{Z}$. Then **OptimisticFTRL**$[R(\cdot), \eta]$ *is defined as follows,*

$$z_t \leftarrow \arg\min_{z \in \mathcal{Z}} \left(\sum_{s=1}^{t-1} \alpha_s \ell_s(z) \right) + \alpha_t m_t(z) + \frac{1}{\eta} R(z), \tag{2.26}$$

where $m_t(\cdot)$ is the hint (or the guess) *for the loss function $\ell_t(\cdot)$; and we assume that each $m_t(\cdot)$ is a $\hat{\mu}_t$-strongly convex function over $\mathcal{Z}$.* OPTIMISTICFTRL *satisfies,*

$$\begin{aligned}
&\boldsymbol{\alpha}\text{-}\mathrm{REG}^{z} \leq && (2.27)\\
&\quad \sum_{t=1}^{T} \alpha_t \left(\ell_t(z_t) - \ell_t(w_{t+1}) - m_t(z_t) + m_t(w_{t+1}) \right) && \text{(term (A))}\\
&\quad + \frac{1}{\eta} \left(R(z^*) - R(w_1) \right) && \text{(term (B))}\\
&\quad - \frac{1}{2} \sum_{t=1}^{T} \left(\frac{\beta}{\eta} + \sum_{s=1}^{t-1} \alpha_s \mu_s \right) \|z_t - w_t\|^2 && \text{(term (C))}\\
&\quad - \frac{1}{2} \sum_{t=1}^{T} \left(\frac{\beta}{\eta} + \alpha_t \hat{\mu}_t + \sum_{s=1}^{t-1} \alpha_s \mu_s \right) \|z_t - w_{t+1}\|^2 && \text{(term (D))}
\end{aligned}$$

where $w_t :=$ argmin$_{z \in \mathcal{Z}} \sum_{s=1}^{t-1} \alpha_s \ell_s(z) + \frac{1}{\eta} R(z)$*, and z^* is the comparator in the regret definition $\boldsymbol{\alpha}\text{-}\mathrm{REG}^{z} := \sum_{t=1}^{T} \alpha_t \ell_t(z_t) - \sum_{t=1}^{T} \alpha_t \ell_t(z^*)$ (similarly to the way we define in Equation* (2.9)*).*

Remark: Note that the regret bound actually holds for any comparator $z^* \in \mathcal{Z}$. In our Fenchel game fomulation, we will take z^* be a minimizer of the optimization problem $\min_{x \in \mathcal{K}} f(x)$ and $\mathcal{Z} \leftarrow \mathcal{K}$. The proof of Lemma 8 is deferred to Section 2.4.10. Next we show how the aforementioned guarantees for **FTL**, **FTL**$^{+}$, **OptimisticFTL**, **FTRL**, and **FTRL**$^{+}$ follow from the above Lemma.

Proof of Lemma 5 on **OptimisticFTL** . Observe that OPTIMISTICFTL is actually OPTIMISTICFTRL when $R(\cdot)$ is a zero function. Therefore, let $R(\cdot) = 0$ in Lemma 8 and drop term (C) and (D) in (2.27) as they are non-positive, we obtain the result.

□

Proof of Lemma 7 on **FTRL$^+$**. Observer that the FTRL$^+$ update is exactly equivalent to OPTIMISTICFTRL with $m_t(\cdot) = \ell_t(\cdot)$. Furthermore, w_{t+1} in Lemma 8 is actually z_t of FTRL$^+$ shown on (2.24). So term (A) and term (D) on (2.27) in Lemma 8 is 0, Therefore, FTRL$^+$ regret satisfies

$$\boldsymbol{\alpha}\text{-REG}^z \leq \tfrac{R(z^*)-R(z_0)}{\eta} - \textstyle\sum_{t=1}^{T} \left(\frac{\mu A_{t-1}}{2} + \frac{\beta}{2\eta}\right) \|z_{t-1} - z_t\|^2. \tag{2.28}$$

□

Proof of Lemma 4 on **FTL$^+$**. Observe that FTL$^+$ is actually FTRL$^+$ with $R(\cdot) = 0$. Therefore, let $R(\cdot) = 0$ and $\beta = 0$ in Equation (2.28), we obtain the regret of FTL$^+$

$$\boldsymbol{\alpha}\text{-REG}^z \leq -\textstyle\sum_{t=1}^{T} \frac{\mu A_{t-1}}{2} \|z_{t-1} - z_t\|^2 \leq 0. \tag{2.29}$$

□

Proof of Lemma 6 on **FTRL**. Observe that FTRL is actually OPTIMISTICFTRL where $m_t(\cdot) = 0 \; \forall t$. Therefore, let $m_t(\cdot) = 0$ in Lemma 8, we obtain the regret of FTRL,

$$\boldsymbol{\alpha}\text{-REG}^z \leq \sum_{t=1}^{T} \alpha_t \left(\ell_t(z_t) - \ell_t(z_{t+1})\right) + \frac{1}{\eta}\left(R(z^*) - R(z_1)\right), \tag{2.30}$$

where we have dropped term (C) and term (D) on (2.27) since they are non-positive, and we also note that w_t in Lemma 8 is the same as z_t here. To continue, we use Lemma 9. Specifically, in Lemma 9, we let $\psi_1(\cdot) \leftarrow \sum_{s=1}^{t-1} \alpha_s \ell_s(\cdot) + \frac{1}{\eta} R(\cdot)$ and $\psi_2(\cdot) \leftarrow \sum_{s=1}^{t} \alpha_s \ell_s(\cdot) + \frac{1}{\eta} R(\cdot)$. Then, we have that $\phi(\cdot) = \alpha_t \ell_t(\cdot)$, $u_1 = z_t$, $u_2 = z_{t+1}$ and that $\sigma = \sum_{s=1}^{t} \alpha_s \mu + \beta$. So by Lemma 9 below, we have that

$$\alpha_t \left(\ell_t(z_t) - \ell_t(z_{t+1})\right) \leq \frac{2\alpha_t^2}{\left(\sum_{s=1}^{t} \alpha_s \mu\right) + \beta} \|\nabla \ell_t(z_t)\|_*^2. \tag{2.31}$$

Combining (2.30) and (2.31) leads to the result.

Lemma 9 (Lemma 5 in [153])**.** *Let $\psi_1(\cdot), \psi_2(\cdot) : \mathcal{Z} \to \mathbb{R}$ be two convex functions defined over a closed and convex domain. Denote $u_1 := \arg\min_{z\in\mathcal{Z}} \psi_1(z)$ and $u_2 := \arg\min_{z\in\mathcal{Z}} \psi_2(z)$. Assume that ψ_2 is σ-strongly convex with respect to a norm $\|\cdot\|$. Define $\phi(\cdot) := \psi_2(\cdot) - \psi_1(\cdot)$. Then,*

$$\|u_1 - u_2\| \leq \frac{2}{\sigma}\|\nabla\phi(u_1)\|_*. \tag{2.32}$$

Furthermore, if $\phi(\cdot)$ is convex, then,

$$0 \leq \phi(u_1) - \phi(u_2) \leq \frac{2}{\sigma}\|\nabla\phi(u_1)\|_*^2. \tag{2.33}$$

□

Proof of Lemma 2 on **FTL** . Observe that FTL is actually FTRL with $R(\cdot) = 0$. Therefore, let $R(\cdot) = 0$ and $\beta = 0$ in Lemma 6, we obtain the result.

□

Next, in Subsections 2.4.7, 2.4.9, and 2.4.8, we go on by presenting three additional online learning algorithms that cannot be captured by the OPTIMISTICFTRL Meta-algorithm.

2.4.7 FTPL (Follow the Perturbed Leader)

One of the most powerful techniques that grew our of online learning is the use of *perturbations* as a type of regularization to obtain vanishing regret guarantees. This idea was first suggested and analyzed by Hannan [122], and later simplified and generalized by Kalai and Vempala [149], who coined the name *Follow the Perturbed Leader* (FTPL). The main idea is to solve the FTL optimization problem with an additional random linear function added to the input, and to select[1] z_t as the expectation of the argmin under this perturbation. More

[1]Technically speaking, the results of [149] only considered linear loss functions and hence their analysis did not require taking averages over the input perturbation. While we will not address computational issues

precisely,

$$z_t := \mathbb{E}_\xi \left[\arg\min_{z\in\mathcal{Z}} \left\{\xi^\top z + \sum_{s=1}^{t-1} \alpha_s \ell_s(z)\right\}\right].$$

Here ξ is some random vector drawn according to an appropriately-chosen distribution and $\ell_s(z)$ is the loss function of the player on round s. Curiously, it was shown in [1] that there is a strong connection between FTPL and FTRL.

2.4.8 BestResp$^+$ (Best Response)

Perhaps the most trivial strategy for a prescient learner is to ignore the history of the ℓ_s's, and simply play the best choice of z_t on the current round. We call this algorithm **BestResp**$^+$, defined as

$$z_t = \operatorname*{argmin}_{z\in\mathcal{Z}} \ell_t(z)\ ; \qquad \textbf{(BestResp}^+\textbf{)}\ . \tag{2.34}$$

Lemma 10. *(**BestResp**$^+$) For any sequence of loss functions $\{\alpha_t\ell_t(\cdot)\}_{t=1}^T$, **BestResp**$^+$ ensures,*

$$\overline{\boldsymbol{\alpha}\text{-REG}}^z := \sum_{t=1}^T \alpha_t\ell_t(z_t) - \min_z \sum_{t=1}^T \alpha_t\ell_t(z) \le 0 \tag{2.35}$$

Proof. Since $z_t = \operatorname{argmin}_{z\in\mathcal{Z}} \ell_t(z)$, we have that $\ell_t(z_t) \le \ell_t(z)$ for any $z \in \mathcal{Z}$. The result follows by summing the inequalities from $t = 1, \dots, T$, and recalling that the α_t's are non-negative. □

2.4.9 OMD$^+$ (Prescient Mirror Descent)

For any sequence of proper lower semi-continuous convex functions $\{\alpha_t\ell_t(\cdot)\}_{t=1}^T$, consider that the player uses **OMD**$^+$ for updating its action, which is defined as follows.

$$z_t := \operatorname{argmin}_{z\in\mathcal{Z}} \alpha_t\ell_t(z) + \tfrac{1}{\gamma} D^\phi_{z_{t-1}}(z)\ ; \qquad (\textbf{OMD}^+\ [\phi(\cdot),\gamma]) \tag{2.36}$$

here due to space, actually computing the average argmin is indeed non-trivial.

where we recall that the Bregman divergence $D_z^\phi(\cdot)$ is with respect to a β-strongly convex distance generating function $\phi(\cdot)$(see Equation (2.3)). Note that in the above definition of OMD$^+$, we assume that the online player is *prescient*, i.e., it knows the loss functions ℓ_t prior to choosing z_t. Recall that in FGNRD (protocol 3) the x-player is allowed to view the current loss prior to playing, and can therefore apply OMD$^+$.

Lemma 11. *(***OMD**$^+$ $[\phi(\cdot), \gamma]$*) Assume that the Bregman Divergence is uniformly bounded on $\mathcal{Z}$, so that $D := D_{z_0}^\phi(z^*)$, where z_0, z^* are any points in $\mathcal{Z}$. For any sequence of proper lower semi-continuous convex loss functions $\{\alpha_t \ell_t(\cdot)\}_{t=1}^T$, the weighted regret of* **OMD**$^+$ $[\phi(\cdot), \gamma]$ *(Equation* (2.36)*) is bounded as follows,*

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^z \leq \frac{D}{\gamma} - \sum_{t=1}^T \frac{\beta}{2\gamma}\|z_{t-1} - z_t\|^2.$$

Proof. The key inequality we need is Lemma 12; using the lemma with $\theta(z) = \gamma(\alpha_t \ell_t(z))$, $z^+ = z_t$ and $c = z_{t-1}$ we have that

$$\gamma(\alpha_t \ell_t(z_t)) - \gamma(\alpha_t \ell_t(z^*)) = \theta(z_t) - \theta(z^*) \leq D_{z_{t-1}}^\phi(z^*) - D_{z_t}^\phi(z^*) - D_{z_{t-1}}^\phi(z_t). \quad (2.37)$$

Therefore, we have that

$$\begin{aligned}
&\boldsymbol{\alpha}\text{-}\mathrm{REG}^z := \textstyle\sum_{t=1}^T \alpha_t \ell_t(z_t) - \min_{z \in \mathcal{X}} \sum_{t=1}^T \alpha_t \ell_t(z) \\
&\overset{(2.37)}{\leq} \textstyle\sum_{t=1}^T \frac{1}{\gamma}\big(D_{z_{t-1}}^\phi(z^*) - D_{z_t}^\phi(z^*) - D_{z_{t-1}}^\phi(z_t)\big) \\
&= \tfrac{1}{\gamma} D_{z_0}^\phi(z^*) - \tfrac{1}{\gamma} D_{z_T}^\phi(z^*) + \textstyle\sum_{t=1}^{T-1} (\frac{1}{\gamma} - \frac{1}{\gamma}) D_{z_t}^\phi(z^*) - \frac{1}{\gamma} D_{z_{t-1}}^\phi(z_t) \\
&= \tfrac{D}{\gamma} - \textstyle\sum_{t=1}^T \frac{\beta}{2\gamma}\|z_{t-1} - z_t\|^2,
\end{aligned} \quad (2.38)$$

where the last inequality uses the definition of D and the strong convexity of ϕ, which grants $D_{z_{t-1}}^\phi(z_t) \geq \frac{\beta}{2}\|z_t - z_{t-1}\|^2$. □

Lemma 12 (Property 1 in [259])**.** *For any proper lower semi-continuous convex function*

$\theta(z)$, *let* $z^+ = \text{argmin}_{z\in\mathcal{Z}}\, \theta(z) + D_c^\phi(z)$. *Then, it satisfies that for any* $z^* \in \mathcal{Z}$,

$$\theta(z^+) - \theta(z^*) \le D_c^\phi(z^*) - D_{z^+}^\phi(z^*) - D_c^\phi(z^+). \tag{2.39}$$

Proof. The result is quite well-known and also appeared in (e.g. [49]). For completeness, we replicate the proof here. Recall that the Bregman divergence with respect to the distance generating function $\phi(\cdot)$ at a point c is: $D_c^\phi(z) := \phi(z) - \langle \nabla\phi(c), z - c\rangle - \phi(c)$.

Denote $F(z) := \theta(z) + D_c^\phi(z)$. Since z^+ is the optimal point of $\min_{z\in\mathcal{Z}} F(z)$, by optimality,

$$\langle z^* - z^+, \nabla F(z^+)\rangle = \langle z^* - z^+, \partial\theta(z^+) + \nabla\phi(z^+) - \nabla\phi(c)\rangle \ge 0, \tag{2.40}$$

for any $z^* \in \mathcal{Z}$. Now using the definition of subgradient, we have that

$$\theta(z^*) \ge \theta(z^+) + \langle \partial\theta(z^+), z^* - z^+\rangle. \tag{2.41}$$

By combining (2.40) and (2.41), we have that

$$\begin{aligned}
\theta(z^*) &\ge \theta(z^+) + \langle \partial\theta(z^+), z^* - z^+\rangle. \\
&\ge \theta(z^+) + \langle z^* - z^+, \nabla\phi(c) - \nabla\phi(z^+)\rangle. \\
&= \theta(z^+) - \{\phi(z^*) - \langle \nabla\phi(c), z^* - c\rangle - \phi(c)\} \\
&+ \{\phi(z^*) - \langle \nabla\phi(z^+), z^* - z^+\rangle - \phi(z^+)\} \\
&+ \{\phi(z^+) - \langle \nabla\phi(c), z^+ - c\rangle - \phi(c)\} \\
&= \theta(z^+) - D_c^\phi(z^*) + D_{z^+}^\phi(z^*) + D_c^\phi(z^+).
\end{aligned} \tag{2.42}$$

□

2.4.10 Proof of Lemma 8

Proof. We can re-write the regret as

$$\begin{aligned}
\boldsymbol{\alpha}\text{-}\mathrm{REG}^z &:= \sum_{t=1}^{T} \alpha_t \ell_t(z_t) - \sum_{t=1}^{T} \alpha_t \ell_t(z^*) \\
&= \underbrace{\sum_{t=1}^{T} \alpha_t \left(\ell_t(z_t) - \ell_t(w_{t+1})\right) - \alpha_t \left(m_t(z_t) - m_t(w_{t+1})\right)}_{\text{first term}} \\
&\quad + \underbrace{\sum_{t=1}^{T} \alpha_t \left(m_t(z_t) - m_t(w_{t+1})\right)}_{\text{second term}} \quad + \underbrace{\sum_{t=1}^{T} \alpha_t \left(\ell_t(w_{t+1}) - \ell_t(z^*)\right)}_{\text{third term}}.
\end{aligned} \tag{2.43}$$

In the following, we will denote

$$D_T := \frac{1}{2}\left(\sum_{t=1}^{T} \big(\frac{\beta}{\eta} + \sum_{s=1}^{t} \alpha_s \mu_s\big)\|z_t - w_t\|^2 + \sum_{t=1}^{T} \big(\frac{\beta}{\eta} + \sum_{s=1}^{t} \alpha_s \mu_s + \alpha_t \hat{\mu}_t\big)\|z_t - w_{t+1}\|^2\right)$$

for brevity. Let us first deal with the second term and the third term. We will use induction to show that

$$\begin{aligned}
&\underbrace{\sum_{t=1}^{T} \alpha_t \left(m_t(z_t) - m_t(w_{t+1})\right)}_{\text{second term}} \quad + \underbrace{\sum_{t=1}^{T} \alpha_t \left(\ell_t(w_{t+1}) - \ell_t(z^*)\right)}_{\text{third term}} \\
&\leq \frac{1}{\eta}\left(R(z^*) - R(w_1)\right) - \left(\sum_{t=1}^{T} \big(\frac{\beta}{\eta} + \sum_{s=1}^{t} \alpha_s \mu_s\big)\left(\|z_t - w_t\|^2 + \|z_t - w_{t+1}\|^2\right)\right),
\end{aligned} \tag{2.44}$$

for any point $z^* \in \mathcal{Z}$.For the base case $T = 0$, we have that

$$\begin{aligned}
&\sum_{t=1}^{0} \alpha_t \left(m_t(z_t) - m_t(w_{t+1})\right) + \sum_{t=1}^{0} \alpha_t \left(\ell_t(w_{t+1}) - \ell_t(z^*)\right) \\
&\quad = 0 \leq \frac{1}{\eta}\left(R(z^*) - R(w_1)\right) - 0,
\end{aligned} \tag{2.45}$$

as $w_1 := \arg\min_{z \in \mathcal{Z}} R(z)$. So the base case trivially holds.

Let us assume that the inequality (2.44) holds for $t = 0, 1, \ldots, T-1$. Now consider round T. We have that

$$\begin{aligned}
&\sum_{t=1}^{T} \alpha_t \left(m_t(z_t) - m_t(w_{t+1}) + \ell_t(w_{t+1}) \right) \\
&\overset{(a)}{\leq} \alpha_T \left(m_T(z_T) - m_T(w_{T+1}) + \ell_T(w_{T+1}) \right) + \frac{1}{\eta} \left(R(w_T) - R(w_1) \right) - D_{T-1} \\
&\quad + \sum_{t=1}^{T-1} \alpha_t \ell_t(w_T). \\
&\overset{(b)}{\leq} \alpha_T \left(m_T(z_T) - m_T(w_{T+1}) + \ell_T(w_{T+1}) \right) + \frac{1}{\eta} \left(R(z_T) - R(w_1) \right) - D_{T-1} \\
&\quad - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s \right) \|z_T - w_T\|^2 + \sum_{t=1}^{T-1} \alpha_t \ell_t(z_T) \\
&= \alpha_T \left(\ell_T(w_{T+1}) - m_T(w_{T+1}) \right) + \frac{1}{\eta} \left(R(z_T) - R(w_1) \right) - D_{T-1} \\
&\quad - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s \right) \|z_T - w_T\|^2 + \sum_{t=1}^{T-1} \alpha_t \ell_t(z_T) + \alpha_T m_T(z_T) \\
&\overset{(c)}{\leq} \alpha_T \left(\ell_T(w_{T+1}) - m_T(w_{T+1}) \right) + \frac{1}{\eta} \left(R(w_{T+1}) - R(w_1) \right) - D_{T-1} \\
&\quad - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s \right) \|z_T - w_T\|^2 - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s \right) \|z_T - w_{T+1}\|^2 \\
&\quad + \sum_{t=1}^{T-1} \alpha_t \ell_t(w_{T+1}) + \alpha_T m_T(w_{T+1}) \\
&= \sum_{t=1}^{T} \alpha_t \ell_t(w_{T+1}) + \frac{1}{\eta} \left(R(w_{T+1}) - R(w_1) \right) - D_T \\
&\overset{(d)}{\leq} \sum_{t=1}^{T} \alpha_t \ell_t(z^*) + \frac{1}{\eta} \left(R(z^*) - R(w_1) \right) - D_T,
\end{aligned} \tag{2.46}$$

where (a) we use the induction such that the inequality (2.44) holds for any $z^* \in \mathcal{Z}$ including $z^* = w_T$, and (b) is because

$$\sum_{t=1}^{T-1} \alpha_t \ell_t(w_T) + \frac{1}{\eta} R(w_T) \leq \sum_{t=1}^{T-1} \alpha_t \ell_t(z_T) + \frac{1}{\eta} R(z_T) - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s \right) \|z_T - w_T\|^2, \tag{2.47}$$

as w_T is the minimizer of a $\frac{\beta}{\eta} + \sum_{t=1}^{T-1} \alpha_t \mu_t$ strongly convex function since

$$w_T := \operatorname*{argmin}_{z \in \mathcal{Z}} \sum_{s=1}^{T-1} \alpha_s \ell_s(z) + \frac{1}{\eta} R(z),$$

and (c) is because

$$\begin{aligned}
&\sum_{t=1}^{T-1} \alpha_t \ell_t(z_T) + \alpha_T m_T(z_T) + \frac{1}{\eta} R(z_T) \\
&\quad \leq \sum_{t=1}^{T-1} \alpha_t \ell_t(w_{T+1}) + \alpha_T m_T(w_{T+1}) + \frac{1}{\eta} R(w_{T+1}) \\
&\qquad - \frac{1}{2} \left(\frac{\beta}{\eta} + \sum_{s=1}^{T-1} \alpha_s \mu_s + \alpha_T \hat{\mu}_T \right) \|z_T - w_{T+1}\|^2,
\end{aligned} \tag{2.48}$$

as z_T is the minimizer of a $\frac{\beta}{\eta} + \sum_{t=1}^{T-1} \alpha_t \mu_t + \alpha_T \hat{\mu}_T$ strongly convex function since

$$z_T := \arg\min_{z \in \mathcal{Z}} \left(\sum_{s=1}^{T-1} \alpha_s \ell_s(z) \right) + \alpha_T m_T(z) + \frac{1}{\eta} R(z),$$

and (d) is due to

$$w_{T+1} := \arg\min_{z \in \mathcal{Z}} \sum_{t=1}^{T} \alpha_t \ell_t(z) + \frac{1}{\eta} R(z).$$

$\square$

2.5 Recovery of existing algorithms

What we are now able to establish, using the tools developed above, is that several iterative first order methods to minimize a convex function can be cast as simple instantiations of the Fenchel game no-regret dynamics. But more importantly, using this framework and the various regret bounds stated above, we able to establish a convergence rate for each via a unified analysis.

For everyone one of the optimization methods we explore below we provide the following:

1. We state the update method described in its standard iterative form, alongside an equivalent formulation given as a no-regret dynamic. To provide the FGNRD form, we must specify the payoff function $g(\cdot, \cdot)$–typically the Fenchel game, with some variants—as well as the sequence of weights α_t, and the no-regret algorithms OAlg^Y, OAlg^X the two players.

2. We provide a proof of this equivalence, showing that the FGNRD formulation does indeed produce the same sequence of iterates as the iterative form; this is often deferred to the appendix.

3. Leaning on Theorem 3, we prove a convergence rate for the method.

2.5.1 Frank-Wolfe method and its variants

The *Frank-Wolfe method* (FW) [91], also known as *conditional gradient*, is known for solving constrained optimization problems. FW is entirely first-order, while requiring access to a linear optimization oracle. Specifically, given a compact and convex constraint set $\mathcal{K} \subset \mathbb{R}^d$, FW relies on the ability to (quickly) answer queries of the form $\text{argmin}_{x \in \mathcal{K}} \langle x, v \rangle$, for any vector $v \in \mathbb{R}^d$. In many cases this linear optimization problem is much faster for well-behaved constraint sets; e.g. simple convex polytopes, the PSD cone, and various balls defined by vector and matrix norms [103, 101, 34]. When the constraint set is the nuclear norm ball, which arises in matrix completion problems, then the linear optimization oracle corresponds to computing a top singular vector, which requires time roughly linear in the size of the matrix [290, 127].

Algorithm 4 Frank-Wolfe [91]

Given: L-smooth $f(\cdot)$, convex domain $\mathcal{K}$, arbitrary w_0, iterations T

$$\begin{aligned} \gamma_t &\leftarrow \tfrac{2}{t+1} \\ v_t &\leftarrow \operatorname*{argmin}_{v \in \mathcal{K}} \langle v, \nabla f(w_{t-1}) \rangle \\ w_t &\leftarrow (1-\gamma_t) w_{t-1} + \gamma_t v_t \end{aligned}$$

Iterative Description

$$\begin{aligned} g(x,y) &:= \langle x, y \rangle - f^*(y) \\ \alpha_t &\leftarrow t \\ \text{OAlg}^Y &:= \text{FTL}[\nabla f(w_0)] \\ \text{OAlg}^X &:= \text{BESTRESP}^+ \end{aligned}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$

We describe the Frank-Wolfe method precisely in Algorithm 4, in both its iterative form and its FGNRD interpretation. We begin by showing that these two representations are equivalent.

Theorem 5. *The two interpretations of Frank-Wolfe, as described in Algorithm 4, are equivalent. That is, for every t, the iterate w_t computed iteratively on the left hand side is identically the weighted-average point $\bar{x}_t$ produced by the dynamic on the right hand side.*

Proof. We show, via induction, that the following three equalities are maintained for every t. Note that three objects on the left correspond to the iterative description given in Algorithm 4 whereas the three on the right correspond to the FGNRD description.

$$\nabla f(w_{t-1}) = y_t \tag{2.49}$$

$$v_t = x_t \tag{2.50}$$

$$w_t = \bar{x}_t. \tag{2.51}$$

To start, we observe that since the OAlg^Y is set as $\text{FTL}[\nabla f(w_0)]$, we have that the base case for (2.49), $y_1 = \nabla f(w_0)$, holds by definition. Furthermore, we observe that for any t we have (2.49) $\implies$ (2.50). This is because, if $y_t = \nabla f(w_{t-1})$, the definition of BESTRESP^+

implies that

$$x_t = \operatorname*{argmin}_{x\in\mathcal{X}} \alpha_t\left(\langle x, y_t\rangle - f^*(y_t)\right) = \operatorname*{argmin}_{x\in\mathcal{X}} \langle x, \nabla f(w_{t-1})\rangle = v_t \tag{2.52}$$

Next, we can show that (2.50) $\implies$ (2.51) for any t as well using induction. Assuming that $w_{t-1} = \frac{\sum_{s=1}^{t-1}\alpha_s x_s}{\sum_{s=1}^{t-1}\alpha_s} = \frac{\sum_{s=1}^{t-1} s v_s}{\sum_{s=1}^{t-1} s}$, a bit of algebra verifies

$$\begin{aligned} w_t &:= (1-\gamma_t)w_{t-1} + \gamma_t v_t = \left(\frac{t-1}{t+1}\right)\frac{\sum_{s=1}^{t-1} s v_s}{\sum_{s=1}^{t-1} s} + \left(\frac{2}{t+1}\right) v_t \\ &= \frac{\sum_{s=1}^{t} s v_s}{\sum_{s=1}^{t} s} = \frac{\sum_{s=1}^{t}\alpha_s x_s}{\sum_{s=1}^{t}\alpha_s} =: \bar{x}_t \end{aligned}$$

Finally, we show that (2.49) holds for $t > 1$ via induction. Recall that y_t is selected via FTL against the sequence of loss functions $\alpha_t \ell_t(\cdot) := -\alpha_t g(x_t, \cdot)$ Precisely this means that, for $t > 1$,

$$\begin{aligned} y_t &:= \arg\min_{y\in\mathcal{Y}} \left\{\tfrac{1}{A_{t-1}}\textstyle\sum_{s=1}^{t-1}\alpha_s \ell_s(y)\right\} \\ &= \arg\min_{y\in\mathcal{Y}} \left\{\frac{1}{A_{t-1}}\sum_{s=1}^{t-1}\alpha_s(-x_s^\top y + f^*(y))\right\} \\ &= \arg\max_{y\in\mathcal{Y}} \left\{\bar{x}_{t-1}^\top y - f^*(y)\right\} = \nabla f(\bar{x}_{t-1}), \end{aligned}$$

The final line follows as a result of the Legendre transform [30]. Finally, by induction, we have that $\bar{x}_{t-1} = w_{t-1}$, and hence we have established (2.49). This completes the proof.

$\square$

Now that we have established Frank-Wolfe as an instance of Protocol 3, we can now prove a bound on convergence using the tools established in Section 2.3.

Theorem 6. *Let w_T be the output of Algorithm 4. Let f be L-smooth and let $\mathcal{K}$ have* *squared ℓ_2 diameter no more than D. Then we have*

$$f(w_T) - \min_{w \in \mathcal{K}} f(w) \leq \frac{8LD}{T+1}.$$

Proof. Now that we have established that Algorithm 4 is an instance of Protocol 3, we can appeal directly to Theorem 3 to see that

$$f(w_T) - \min_{w \in \mathcal{K}} f(w) \leq \overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}}^x[\mathrm{BESTRESP}^+] + \overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}}^y[\mathrm{FTL}].$$

Recall that, by Lemma 10, we have that $\overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}}^x[\mathrm{BESTRESP}^+] \leq 0$. Let us then turn our attention to the regret of OAlg^Y.

First note that, since $f(\cdot)$ is L-smooth, its conjugate $f^*(\cdot)$ is $\frac{1}{L}$-strongly convex, and thus the function $-g(x, \cdot)$ is also $\frac{1}{L}$-strongly convex in its second argument. Next, if we define $\ell_t(\cdot) := -g(x_t, \cdot)$, then we can bound the norm of the gradient as

$$\|\nabla \ell_t(y_t)\|^2 = \|x_t - \nabla f^*(y_t)\|^2 = \|x_t - \bar{x}_{t-1}\|^2 \leq D.$$

Combining with Lemma 2 we see that

$$\overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}}^y[\mathrm{FTL}] \leq \frac{1}{A_T} \sum_{t=1}^{T} \frac{2\alpha_t^2 \|\nabla \ell_t(y_t)\|^2}{\sum_{s=1}^{t} \alpha_s (1/L)} \quad = \frac{8L}{T(T+1)} \sum_{t=1}^{T} \frac{t^2 D}{t(t+1)} \leq \frac{8LD}{T+1}.$$

This completes the proof.

□

2.5.1.1 Variant 1: a linear rate Frank-Wolfe over strongly convex set

Levitin and Polyak [169], Demyanov and Rubinov [69], Dunn [81] show that under certain conditions, Frank-Wolfe for smooth convex function (*not necessarily a strongly convex function*) for strongly convex sets has linear rate under certain conditions. We show that a similar result can be derived from the game framework.

Theorem 7. *Suppose that* $\min_{x\in\mathcal{K}} f(x)$ *is L-smooth convex. and that* $\mathcal{K}$ *is a* λ*-strongly convex set. Also assume that the gradients of the* $f(\cdot)$ *in* $\mathcal{K}$ *are bounded away from* 0*, i.e.,* $\max_{x\in\mathcal{K}} \|\nabla f(x)\| \geq B$*. Then, there exists a FW-like algorithm that has* $O(\exp(-\frac{\lambda B}{L}T))$ *rate which is an instance of Algorithm 3 with the weighting scheme* $\alpha_t := \frac{1}{\|\nabla \ell_t(y_t)\|^2}$ *if Alg. 3 sets OAlg*^{Y} $:=$ FTL *and OAlg*^{X} $:=$ BESTRESP$^+$.

Note that the weights α_t are not predefined but rather depend on the queries of the algorithm. The proof of Theorem 7 is described in full detail in Section 2.7.2.

2.5.1.2 Variant 2: a smoothing Frank-Wolfe for non-smooth functions

Looking carefully at the proof of Theorem 6, the fact that FTL was suitable for the vanilla FW analysis relies heavily on the strong convexity of the functions $\ell_t(\cdot) := -g(x_t, y)$, which in turn results from the smoothness of $f(\cdot)$. But what about when $f(\cdot)$ is not smooth, is there an alternative algorithm available?

We observe that one of the nice techniques to grow out of the online learning community is the use of *perturbations* as a type of regularization to obtain vanishing regret guarantees [149] – their method is known as *Follow the Perturbed Leader* (FTPL). The main idea is to solve an optimization problem that has a random linear function added to the input, and to select[2] as x_t the expectation of the argmin under this perturbation. More precisely,

$$y_t := \mathbb{E}_Z \left[\arg\min_{y\in Y} \left\{ Z^\top y + \textstyle\sum_{s=1}^{t-1} \ell_s(y) \right\}\right].$$

Here Z is some random vector drawn according to an appropriately-chosen distribution and $\ell_s(x)$ is the loss function of the x-player on round s; with the definition of payoff function g, i.e. $\ell_s(y) := -x_s^\top y + f^*(y)$.

[2] Technically speaking, the results of Kalai and Vempala [149] only considered linear loss functions and hence their analysis did not require taking averages over the input perturbation. While we will not address computational issues here due to space, actually computing the average argmin is indeed non-trivial.

One can show that, as long as Z is chosen from the right distribution, then this algorithm guarantees average regret on the order of $O\left(\frac{1}{\sqrt{T}}\right)$, although obtaining the correct dimension dependence relies on careful probabilistic analysis. Recent work of Abernethy et al. [1] shows that the analysis of perturbation-style algorithm reduces to curvature properties of a stochastically-smoothed Fenchel conjugate.

What is intriguing about this perturbation approach is that it ends up being equivalent to an existing method proposed by Lan [159] (Section 3.3), who also uses a stochastically smoothed objective function. We note that

$$\begin{aligned}
&\mathbb{E}_Z\left[\arg\min_{x\in X}\left\{Z^\top x+\sum_{s=1}^{t-1}\ell_s(x)\right\}\right] \\
&=\mathbb{E}_Z\left[\arg\max_{x\in X}\left\{(\bar{y}_{t-1}+Z/(t-1))^\top x-f^*(x)\right\}\right] \\
&=\mathbb{E}_Z[\nabla f(\bar{y}_{t-1}+Z/(t-1))]=\nabla\tilde{f}_{t-1}(\bar{y}_{t-1})
\end{aligned} \tag{2.53}$$

where $\tilde{f}_\alpha(x) := \mathbb{E}[f(x+Z/\alpha)]$. Lan [159] suggests using precisely this modified $\tilde{f}$, and they prove a rate on the order of $O\left(\frac{1}{\sqrt{T}}\right)$. As discussed, the same would follow from vanishing regret of FTPL. In other words, by plugging in FTPL as the alternative algorithm, what we're actually doing is using a "stochastically smoothed" version of f.

2.5.1.3 Variant 3: an incremental Frank-Wolfe

Recently, Négiar et al. [203] and Lu and Freund [186] propose stochastic Frank-Wolfe algorithms for optimizing smooth convex finite-sum functions, i.e. $\min_{x\in\mathcal{K}} f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)$, where each $f_i(x) := \phi(x^\top z_i)$ represents a loss function $\phi(\cdot)$ associated with sample z_i. In each iteration the algorithms only require a gradient computation of a single component, see option (A) of Algorithm 5. Négiar et al. [203] show that the algorithm has $O(\frac{c_\kappa}{T})$ expected convergence rate, where c_κ is a number that depends on the underlying data matrix z and in worst case is bounded by the number of components n. We show that a similar algorithm, option (B) of Algorithm 5, can be generated from Algorithm 3 that has $\tilde{O}(\frac{n}{T})$

deterministic convergence rate, which picks a sample in each iteration by cycling through the data points. We have the following theorem and its proof is in Section 2.7.3.

Algorithm 5 Stochastic Frank-Wolfe algorithm: (option (A) is the algorithm of [203], while option (B) is the algorithm analyzed in this work.)

1: **Init:** $w_0 \in \mathcal{K}$.
2: For each sample i, compute $g_{i,0} := \frac{1}{n}\nabla f_i(w_0) \in \mathbb{R}^d$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Option (A): Sample a $i_t \in [n]$ uniformly at random.
5: Option (B): Select a sample $i_t \in [n]$ by cycling through the samples.
6: Compute $\nabla f_{i_t}(w_t)$ and set $g_{i_t,t} := \frac{1}{n}\nabla f_{i_t}(w_t)$. For other $j \neq i \in [n]$, $g_{j,t} = g_{j,t-1}$.
7: $g_t = \sum_{i=1}^n g_{i,t}$.
8: $v_t = \arg\min_{x \in \mathcal{K}} \langle x, g_t \rangle$.
9: Option (A): $w_t = (1 - \frac{2}{t+1})w_{t-1} + \frac{2}{t+1}v_t$.
10: Option (B): $w_t = (1 - \frac{1}{t})w_{t-1} + \frac{1}{t}v_t$.
11: **end for**
12: Output w_t.

Theorem 8. *When both are run for exactly T rounds, the output $\bar{x}_T$ of Algorithm 3 with the weighting scheme $\{\alpha_t = t\}$ is identically the output w_T of Algorithm 5 with learning rate $\gamma_t = \frac{1}{t}$ as long as:* ***(I)*** *Alg. 3 sets OAlg^Y $:= g_t$ (line 7 of Algorithm 5);* ***(II)*** *Alg. 3 sets OAlg^X $:=$* BESTRESP$^+$. *Furthermore, assume that $f(\cdot)$ is L-smooth convex and that its conjugate is L_0-Lipschitz. Then option (B) of Algorithm 5 outputs w_T with approximation error $O\left(\frac{\max\{LD, L(L_0+r)nr\}\log T}{T}\right)$, where r is a bound of the length of any point x in the constraint set $\mathcal{K}$, i.e. $\max_{x\in\mathcal{K}} \|x\| \leq r$, and D is the squared of the diameter of $\mathcal{K}$.*

2.5.1.4 Related works

Bach [20] shows that for certain types of objectives, subgradient descent applied to the primal domain is equivalent to FW applied to the dual domain. Garber and Hazan [100] shows that for strongly convex and smooth objective functions, FW can achieve $O(1/T^2)$ convergence rate over strongly convex set . Garber and Hazan [99], Garber and Hazan [101] show that exponential convergence for strongly convex and smooth objectives over some polytopes can be achieved by a projection-free algorithm. Their algorithms require a stronger oracle by using the standard one, but can be efficiently implemented for certain polytopes like simplex. Other linear rate of FW-like algrorithms for certain convex polytopes includes [103, 276, 111, 157, 93]. There are also many works of Frank-Wolfe on different aspects, e.g. online learning setting [128], minimizing some structural norms [124, 287], reducing the number of gradient evaluations [161], block-wise update for structural SVM [158, 217, 270]. Finally, we note that Frank-Wolfe has a nice property that it tends to produce sparse solution (see e.g. [139, 56]), as it adds one component at a time.

2.5.2 Accelerated methods for smooth convex optimization

In this subsection, we are going to introduce several accelerated algorithms. To achieve acceleration, we will consider that the y-player in the game plays OPTIMISTICFTL

$$\widetilde{y}_t \leftarrow \arg\min_{y \in \mathcal{Y}} \alpha_t m_t(y) + \sum_{s=1}^{t-1} \alpha_s \ell_s(y), \quad \text{and let} \quad m_t(\cdot) := \ell_{t-1}(\cdot), \tag{2.54}$$

where the learner uses the loss function of the previous round $\ell_{t-1}(\cdot)$ as the guess $m_t(\cdot)$ of the loss function at t before observing the loss function.

For the time being, let us assume that the sequence of x_t's is arbitrary. We define

$$\bar{x}_t := \tfrac{1}{A_t} \textstyle\sum_{s=1}^{t} \alpha_s x_s \qquad \text{and} \qquad \widetilde{x}_t := \tfrac{1}{A_t}(\alpha_t x_{t-1} + \textstyle\sum_{s=1}^{t-1} \alpha_s x_s). \tag{2.55}$$

It is critical that we have two parallel sequences of iterate averages for the x-player. Our final algorithm will output $\bar{x}_T$, whereas the Fenchel game dynamics will involve computing ∇f at the *reweighted averages* $\widetilde{x}_t$ for each $t = 1, \ldots, T$.

To prove the key regret bound for the y-player, we first need to state some simple technical facts.

$$\begin{aligned}
\hat{y}_{t+1} &= \operatorname*{argmin}_y \sum_{s=1}^{t} \alpha_s \left(f^*(y) - \langle x_s, y \rangle \right) = \operatorname*{argmax}_y \langle \bar{x}_t, y \rangle - f^*(y) && (2.56) \\
&= \nabla f(\bar{x}_t) && (2.57) \\
\widetilde{y}_t &= \operatorname*{argmin}_y \alpha_t \left(f^*(y) - \langle x_t, y \rangle \right) + \sum_{s=1}^{t-1} \alpha_s \left(f^*(y) - \langle x_s, y \rangle \right) && (2.58) \\
&= \nabla f(\widetilde{x}_t), && (2.59) \\
\widetilde{x}_t - \bar{x}_t &= \frac{\alpha_t}{A_t}(x_{t-1} - x_t). && (2.60)
\end{aligned}$$

Equations (2.56) and (2.58) follow from elementary properties of Fenchel conjugation and the Legendre transform [232]. Equation (2.60) follows from a simple algebraic calculation.

Lemma 13. *Suppose $f(\cdot)$ is a convex function that is L-smooth with respect to the the norm $\|\cdot\|$ with dual norm $\|\cdot\|_*$. Let $x_1, \ldots, x_T$ be an arbitrary sequence of points. Then, we have*

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^y(\widetilde{y}_1, \ldots, \widetilde{y}_T) \leq L \textstyle\sum_{t=1}^{T} \frac{\alpha_t^2}{A_t} \|x_{t-1} - x_t\|^2. \tag{2.61}$$

Proof. Using Lemma 5 with $m_t(\cdot) \leftarrow \ell_{t-1}(\cdot)$, $w_t \leftarrow \hat{y}_t$, and $z_t \leftarrow \tilde{y}_t$, and that $\alpha_t \left(\ell_t(y) - \ell_{t-1}(y) \right)$ $= \alpha_t \langle x_{t-1} - x_t, y \rangle$ in Fenchel Game, we have

$$\begin{aligned}
\textstyle\sum_{t=1}^{T} \alpha_t \ell_t(\widetilde{y}_t) - \alpha_t \ell_t(y^*) &\leq \textstyle\sum_{t=1}^{T} \alpha_t \left(\ell_t(\widetilde{y}_t) - \ell_{t-1}(\widetilde{y}_t) - \left(\ell_t(\hat{y}_{t+1}) - \ell_{t-1}(\hat{y}_{t+1}) \right) \right) \\
\text{(Eqns. 2.56, 2.58)} \qquad &= \textstyle\sum_{t=1}^{T} \alpha_t \langle x_{t-1} - x_t, \nabla f(\widetilde{x}_t) - \nabla f(\bar{x}_t) \rangle \\
\text{(Hölder's Ineq.)} \qquad &\leq \textstyle\sum_{t=1}^{T} \alpha_t \|x_{t-1} - x_t\| \|\nabla f(\widetilde{x}_t) - \nabla f(\bar{x}_t)\|_* \\
\text{(L-smoothness of f)} \qquad &\leq L \textstyle\sum_{t=1}^{T} \alpha_t \|x_{t-1} - x_t\| \|\widetilde{x}_t - \bar{x}_t\| \\
\text{(Eqn. 2.60)} \qquad &= L \textstyle\sum_{t=1}^{T} \frac{\alpha_t^2}{A_t} \|x_{t-1} - x_t\| \|x_{t-1} - x_t\|
\end{aligned}$$

as desired, where the first inequality is because that $m_t(\cdot) = \ell_{t-1}(\cdot)$. □

Theorem 9. *Let us consider the output $(\bar{x}_T, \bar{y}_T)$ of Algorithm 3 under the following conditions: (a) the sequence $\{\alpha_t\}$ is positive but otherwise arbitrary (b) OAlg^y is chosen*

OPTIMISTICFTL, *(c) OAlg^x is* OMD$^+$ *with a parameter γ, and (d) we have a bound $V_{x_0}(x^*) \leq D$. Then the point $\bar{x}_T$ satisfies*

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{1}{A_T} \left(\frac{D}{\gamma} + \sum_{t=1}^{T} \left(\frac{\alpha_t^2}{A_t} L - \frac{\beta}{2\gamma} \right) \|x_{t-1} - x_t\|^2 \right). \tag{2.62}$$

On the other hand, following the same setting, if OAlg^x is chosen as FTRL$^+$ *with a β-strongly convex regularizer $R(\cdot)$ and a parameter η. Then the point $\bar{x}_T$ satisfies*

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{1}{A_T} \left(\frac{R(x^*) - R(\hat{x})}{\eta} + \sum_{t=1}^{T} \left(\frac{\alpha_t^2}{A_t} L - \frac{\beta}{2\eta} \right) \|x_{t-1} - x_t\|^2 \right), \tag{2.63}$$

where $R(\hat{x}) = \min_{x \in \mathcal{X}} R(x)$.

Proof. We have already done the hard work to prove this theorem. Lemma 1 tells us we can bound the error of $\bar{x}_T$ by the ϵ error of the approximate equilibrium $(\bar{x}_T, \bar{y}_T)$. Theorem 3 tells us that the pair $(\bar{x}_T, \bar{y}_T)$ derived from Algorithm 3 is controlled by the sum of averaged regrets of both players, $\frac{1}{A_T}(\boldsymbol{\alpha}\text{-}\mathrm{REG}^x[\mathrm{OMD}^+] + \boldsymbol{\alpha}\text{-}\mathrm{REG}^y[\mathrm{OPTIMISTICFTL}])$. But we now have control over both of these two regret quantities, from Lemmas 13 of OPTIMISTICFTL and 11 of OMD$^+$,

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{1}{A_T} \left(\frac{D}{\gamma} + \sum_{t=1}^{T} \left(\frac{\alpha_t^2}{A_t} L - \frac{\beta}{2\gamma} \right) \|x_{t-1} - x_t\|^2 \right). \tag{2.64}$$

On the other hand, if the y-player is OPTIMISTICFTL and the x-player is FTRL$^+$, then, by Lemma 13 of OPTIMISTICFTL and Lemma 7 of FTRL$^+$ with $\mu = 0$ (as the x-player sees linear loss functions), we have

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{1}{A_T} \left(\frac{R(x^*) - R(\hat{x})}{\eta} + \sum_{t=1}^{T} \left(\frac{\alpha_t^2}{A_t} L - \frac{\beta}{2\eta} \right) \|x_{t-1} - x_t\|^2 \right), \tag{2.65}$$

where $R(\hat{x}) = \min_{x \in \mathcal{X}} R(x)$.

$\square$

Theorem 3 is somewhat opaque without a specifying the sequence $\{\alpha_t\}$. But what we now show is that the summation term *vanishes* when we can guarantee that $\frac{\alpha_t^2}{A_t}$ remains constant! This is where we obtain the following fast rate.

Corollary 1. *Following the setting as Theorem 9, if the x-player is MD with a 1-strongly convex distance generating function $\phi(\cdot)$ and the parameter γ that satisfies $\frac{1}{CL} \leq \gamma \leq \frac{1}{4L}$ for some constant $C \geq 4$, then*

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{2CLD}{T^2},$$

where $D_{x_0}^{\phi}(x^*) \leq D$. Similarly, if the x-player is FTRL^+ *with a 1-strongly convex regularizer $R(\cdot)$, and the parameter γ satisfies $\frac{1}{CL} \leq \eta \leq \frac{1}{4L}$ for some constant $C \geq 4$ then*

$$f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \leq \frac{2CL\big(R(x^*) - R(\hat{x})\big)}{T^2},$$

where $R(\hat{x}) = \min_{x \in \mathcal{X}} R(x)$.

Proof. As we use $\alpha_t = t$, we have that $A_t := \frac{t(t+1)}{2}$. The choice of $\{\alpha_t, \gamma\}$ implies $\frac{D}{\gamma} \leq CLD$ and $\frac{L\alpha_t^2}{A_t} = \frac{2Lt^2}{t(t+1)} \leq 2L \leq \frac{1}{2\gamma}$, which ensures that the summation term in (2.62) is negative. The rest is simple algebra.

Similar calculations can be done for the bound (2.63), and hence omitted. $\square$

It is worth dwelling on exactly how we obtained the above result. A less refined analysis of the OMD^+ algorithm would have simply ignored the negative summation term in Lemma 11, and simply upper bounded this by 0. But the negative terms $\|x_t - x_{t-1}\|^2$ in this sum happen to correspond *exactly* to the positive terms one obtains in the regret bound for the y-player, but this is true *only as a result of* using the OPTIMISTICFTL algorithm. To obtain a cancellation of these terms, we need a γ_t which is roughly constant, and hence we need to ensure that $\frac{\alpha_t^2}{A_t} = O(1)$. The final bound, of course, is determined by the inverse quantity $\frac{1}{A_T}$, and a quick inspection reveals that the best choice of $\alpha_t = \theta(t)$. This is not the only choice that could work, and we conjecture that there are scenarios in which better bounds are achievable for different α_t tuning. We show in Subsection 2.5.3 that a *linear rate* is achievable when $f(\cdot)$ is also strongly convex, and there we tune α_t to grow exponentially in t rather than linearly.

2.5.2.1 Nesterov's methods

Algorithm 6 Nesterov's 1-memory method [206, 259]

Given: L-smooth $f(\cdot)$, convex domain $\mathcal{K}$, arbitrary $v_0 \in \mathcal{K}$, 1-strongly convex distance generating function $\phi(\cdot)$, iterations T

$$\begin{aligned} \beta_t &\leftarrow \tfrac{2}{t+1}, \gamma_t \leftarrow \tfrac{t}{4L} \\ z_t &\leftarrow (1-\beta_t)w_{t-1} + \beta_t v_{t-1} \\ v_t &\leftarrow \operatorname*{argmin}_{x\in\mathcal{K}} \gamma_t \langle \nabla f(z_t), x\rangle + D^{\phi}_{v_{t-1}}(x) \\ w_t &\leftarrow (1-\beta_t)w_{t-1} + \beta_t v_t \end{aligned}$$

Iterative Description

$$\begin{aligned} g(x,y) &:= \langle x, y\rangle - f^*(y) \\ \alpha_t &:= t \text{ for } t = 1,\dots,T \\ \text{OAlg}^Y &:= \textsc{OptimisticFTL}[\nabla f(v_0)] \\ \text{OAlg}^X &:= \text{OMD}^+[\phi(\cdot), \tfrac{1}{4L}] \end{aligned}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$,

Algorithm 7 Nesterov's ∞-memory method [208, 259]

Given: L-smooth $f(\cdot)$, convex domain $\mathcal{K}$, arbitrary $v_0 \in \mathcal{K}$, 1-strongly convex regularizer $R(\cdot)$, iterations T

$$\begin{aligned} \beta_t &\leftarrow \tfrac{2}{t+1}, \gamma_t \leftarrow \tfrac{t}{4L} \\ z_t &\leftarrow (1-\beta_t)w_{t-1} + \beta_t v_{t-1} \\ v_t &\leftarrow \operatorname*{argmin}_{x\in\mathcal{K}} \sum_{s=1}^{t} \gamma_s \langle \nabla f(z_s), x\rangle + R(x) \\ w_t &\leftarrow (1-\beta_t)w_{t-1} + \beta_t v_t \end{aligned}$$

Iterative Description

$$\begin{aligned} g(x,y) &:= \langle x, y\rangle - f^*(y) \\ \alpha_t &:= t \text{ for } t = 1,\dots,T \\ \text{OAlg}^Y &:= \textsc{OptimisticFTL}[\nabla f(v_0)] \\ \text{OAlg}^X &:= \text{FTRL}^+[R(\cdot), \tfrac{1}{4L}] \end{aligned}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$,

Starting from 1983, Nesterov has proposed three accelerated methods for smooth convex problems (i.e. [205, 204, 206, 208]). In this section, we show that our accelerated algorithm to the *Fenchel game* can generate all the methods with some simple tweaks.

We first consider recovering Nesterov's (1988) 1-memory method [206] and Nesterov's (2005) ∞-memory method [208]. To be precise, we adopt the presentation of Nesterov's algorithm given in Algorithm 1 and Algorithm 3 of Tseng [259] respectively.

Theorem 10. *The two interpretations of Nesterov's* 1*-memory method (Nesterov's* ∞*-memory method, as described in Algorithm 6 (Algorithm 7, respectively), are equivalent. That is, for every* t*, the iterate* w_t *computed iteratively on the left hand side is identically the weighted-average point* $\bar{x}_t$ *produced by the dynamic on the right hand side.*

Proof. Let us recall the notations (2.55), $\bar{x}_t := \frac{1}{A_t}\sum_{s=1}^{t}\alpha_s x_s$ and $\widetilde{x}_t := \frac{1}{A_t}(\alpha_t x_{t-1} + \sum_{s=1}^{t-1}\alpha_s x_s)$. We show, via induction, that the following three equalities are maintained for every t. Note that three objects on the left correspond to the iterative description given in Algorithm 6 whereas the three on the right correspond to the FGNRD description.

$$\nabla f(z_t) = y_t \tag{2.66}$$

$$v_t = x_t \tag{2.67}$$

$$w_t = \bar{x}_t. \tag{2.68}$$

We first note that the initialization ensures that (2.66) holds for $t = 1$. Second, the choices of learning rate γ_t and the weighting scheme $\{\alpha_t\}$ leads to

$$w_t = \frac{1}{\sum_{s=1}^{t} s}\sum_{s=1}^{t} s v_s \;=\; \frac{1}{A_t}\sum_{s=1}^{t}\alpha_s v_s, \quad \text{if } (\beta_t = \frac{2}{t+1}, \alpha_t = t). \tag{2.69}$$

From (2.69), we see that (2.67) implies (2.68), as w_t is always an average of the updates v_t. It remains to establish (2.66) and (2.67) via induction.

Let us first show (2.66). We have already shown in (2.58) that $y_t = \nabla f(\tilde{x}_t)$. So it suffices to show that $\tilde{x}_t = z_t$. We have that $z_t = (1-\beta_t)w_{t-1} + \beta_t x_{t-1} = (1-\beta_t)(\sum_{s=1}^{t-1}\frac{\alpha_s}{A_{t-1}}x_s) + \beta_t x_{t-1} = (1-\frac{2}{t+1})(\sum_{t=1}^{t-1}\frac{\alpha_t}{\frac{t(t-1)}{2}}x_t) + \beta_t x_{t-1} = \sum_{s=1}^{t-1}\frac{\alpha_s}{\frac{t(t+1)}{2}}x_s + \beta_t x_{t-1} = \sum_{s=1}^{t-1}\frac{\alpha_s}{A_t}x_s + \frac{\alpha_t}{A_t}x_{t-1} = \widetilde{x}_t$. To show (2.67), observe that the update on line 5 of Algorithm 3 is exactly equivalent to OMD$^+$ shown on (2.36) for which $\gamma \leftarrow \frac{1}{4L}$ and $\theta_t \leftarrow y_t = \nabla f(z_t)$.

Since by induction, $x_{t-1} = v_{t-1}$, we have that $x_t = v_t$. We thus have completed the first part of proof.

Similar analysis can be conducted for the equivalency between Nesterov's ∞-memory method. Specifically, the method corresponds to FTRL$^+$ is used as the x-player's strategy. □

2.5.2.2 Acceleration for unconstrained smooth convex problems

Algorithm 8 Nesterov's first acceleration method [204, 205]

Given: L-smooth $f(\cdot)$, arbitrary $z_0 \in \mathbb{R}^d$, iterations T

$$\begin{array}{rl} \theta & \leftarrow \frac{t}{2(t+1)L}, \beta_t \leftarrow \frac{t-1}{t+2} \\ w_t & \leftarrow z_{t-1} - \theta \nabla f(z_{t-1}) \\ z_t & \leftarrow w_t + \beta_t(w_t - w_{t-1}) \end{array}$$

Iterative Description

$$\begin{array}{rl} g(x,y) & := \langle x, y\rangle - f^*(y) \\ \alpha_t & := t \text{ for } t = 1, \ldots, T \\ \text{OAlg}^Y & := \text{OPTIMISTICFTL}[\nabla f(z_0)] \\ \text{OAlg}^X & := \text{OMD}^+[\frac{1}{2}\|\cdot\|_2^2, \frac{1}{4L}] \end{array}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$

Now let us consider that the x-player's action space is unconstrained. That is, $\mathcal{K} = \mathbb{R}^n$. We are going to show that our framework can recover Nesterov's first acceleration method [205, 204] (see also [248]).

Theorem 11. *The interpretations of Nesterov's first acceleration method [205, 204] as described in Algorithm 8 are equivalent. That is, for every t, the iterate w_t computed iteratively on the left hand side is identically the weighted-average point $\bar{x}_t$ produced by the dynamic on the right hand side.*

Proof. First of all, in the OMD$^+$ strategy of the x-player, we can let the distance generating function of the Bregman divergence to be the squared of L2 norm, i.e. $\phi(x) := \frac{1}{2}\|x\|_2^2$. Then, the update becomes $x_t = \operatorname{argmin}_x \gamma_t\langle x, \alpha_t y_t\rangle + V_{x_{t-1}}(x) = \operatorname{argmin}_x \gamma_t\langle x, \alpha_t y_t\rangle + \frac{1}{2}\|x\|_2^2 - \langle x_{t-1}, x - x_{t-1}\rangle - \frac{1}{2}\|x_{t-1}\|_2^2$. Differentiating the objective w.r.t x and setting it to zero, one will get $x_t = x_{t-1} - \gamma_t\alpha_t y_t$.

To see the equivalence, let us re-write $\bar{x}_t := \frac{1}{A_t}\sum_{s=1}^{t}\alpha_s x_s$ as follows,

$$\begin{aligned}
\bar{x}_t &= \frac{A_{t-1}\bar{x}_{t-1} + \alpha_t x_t}{A_t} = \frac{A_{t-1}\bar{x}_{t-1} + \alpha_t(x_{t-1} - \gamma_t\alpha_t\nabla f(\widetilde{x}_t))}{A_t} \\
&= \frac{A_{t-1}\bar{x}_{t-1} + \alpha_t(\frac{A_{t-1}\bar{x}_{t-1} - A_{t-2}\bar{x}_{t-2}}{\alpha_{t-1}} - \gamma_t\alpha_t\nabla f(\widetilde{x}_t))}{A_t} \\
&= \bar{x}_{t-1}(\frac{A_{t-1}}{A_t} + \frac{\alpha_t(\alpha_{t-1} + A_{t-2})}{A_t\alpha_{t-1}}) - \bar{x}_{t-2}(\frac{\alpha_t A_{t-2}}{A_t\alpha_{t-1}}) - \frac{\gamma_t\alpha_t^2}{A_t}\nabla f(\widetilde{x}_t) \\
&= \bar{x}_{t-1} - \frac{\gamma_t\alpha_t^2}{A_t}\nabla f(\widetilde{x}_t) + (\frac{\alpha_t A_{t-2}}{A_t\alpha_{t-1}})(\bar{x}_{t-1} - \bar{x}_{t-2}) \\
&= \bar{x}_{t-1} - \frac{t}{2(t+1)L}\nabla f(\widetilde{x}_t) + (\frac{t-2}{t+1})(\bar{x}_{t-1} - \bar{x}_{t-2}).
\end{aligned} \tag{2.70}$$

□

Let us switch to comparing the update of (2.70) of Nesterov's method with the update of the HEAVYBALL algorithm. We see that (2.70) has the so called momentum term (i.e. has a $(\bar{x}_{t-1} - \bar{x}_{t-2})$ term). But, the difference is that the gradient is evaluated at $\widetilde{x}_t = \frac{1}{A_t}(\alpha_t x_{t-1} + \sum_{s=1}^{t-1}\alpha_s x_s)$, not $\bar{x}_{t-1} = \frac{1}{A_{t-1}}\sum_{s=1}^{t-1}\alpha_s x_s$, which is the consequence that the y-player plays OPTIMISTICFTL. To elaborate, let us consider a scenario (shown in Algorithm 9) such that the y-player plays FTL instead of OPTIMISTICFTL.

Algorithm 9 Heavy Ball

Given: L-smooth $f(\cdot)$, arbitrary $z_0 \in \mathcal{K}$, iterations T

$$\begin{aligned}
\eta_t &\leftarrow \tfrac{t}{2(t+1)L}, \quad \beta_t \leftarrow \tfrac{t-1}{t+2} \\
v_t &\leftarrow w_{t-1} - w_{t-2} \\
w_t &\leftarrow w_{t-1} - \eta_t\nabla f(w_{t-1}) + \beta_t v_t
\end{aligned}$$

Iterative Description

$$\begin{aligned}
g(x,y) &:= \langle x, y\rangle - f^*(y) \\
\alpha_t &:= t \text{ for } t = 1,\ldots,T \\
\text{OAlg}^Y &:= \text{FTL}[\nabla f(w_0)] \\
\text{OAlg}^X &:= \text{OMD}^+[\tfrac{1}{2}\|\cdot\|_2^2, \tfrac{1}{4L}]
\end{aligned}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$

Following what we did in (2.70), we can rewrite $\bar{x}_t$ of Algorithm 9 as

$$\bar{x}_t = \bar{x}_{t-1} - \frac{\gamma_t\alpha_t^2}{A_t}\nabla f(\bar{x}_{t-1}) + (\bar{x}_{t-1} - \bar{x}_{t-2})(\frac{\alpha_t A_{t-2}}{A_t\alpha_{t-1}}), \tag{2.71}$$

by observing that (2.70) still holds except that $\nabla f(\widetilde{x}_t)$ is changed to $\nabla f(\bar{x}_{t-1})$ as the y-player uses FTL now, which give us the update of the Heavy Ball algorithm as (2.71). Moreover, by the regret analysis, we have the following theorem. The proof is in Section 2.7.4.

Theorem 12. *Let $\alpha_t = t$. Assume $\mathcal{K} = \mathbb{R}^n$. Also, let $\gamma_t = O(\frac{1}{L})$. The output $\bar{x}_T$ of Algorithm 9 is an $O(\frac{1}{T})$-approximate optimal solution of* $\min_x f(x)$.

To conclude, by comparing Algorithm 8 and Algorithm 9, we see that Nesterov's (1983) method enjoys $O(1/T^2)$ rate since its adopts OPTIMISTICFTL, while the HEAVYBALL algorithm which adopts FTL may not enjoy the fast rate, as the distance terms may not cancel out. The result also conforms to empirical studies that the HEAVYBALL does not exhibit acceleration on general smooth convex problems.

2.5.2.3 *Accelerated proximal method*

Algorithm 10 Accelerated proximal method

Given: L-smooth $f(\cdot)$, arbitrary $w_0 \in \mathbb{R}^d$, iterations T.

$$\begin{array}{rl} \beta_t & \leftarrow \frac{2}{t+1}, \gamma_t \leftarrow \frac{t}{4L} \\ z_t & \leftarrow (1-\beta_t) w_{t-1} + \beta_t v_{t-1} \\ v_t & \leftarrow \mathbf{prox}_{t\gamma\psi}(x_{t-1} - t\gamma \nabla f(z_t)) \\ w_t & \leftarrow (1-\beta_t) w_{t-1} + \beta_t v_t \end{array}$$

Iterative Description

$$\begin{array}{rl} g(x,y) & := \langle x, y\rangle - f^*(y) + \psi(x) \\ \alpha_t & := t \text{ for } t = 1, \dots, T \\ \text{OAlg}^Y & := \text{OPTIMISTICFTL}[\nabla f(v_0)] \\ \text{OAlg}^X & := \text{OMD}^+[\phi(\cdot), \frac{1}{4L}] \end{array}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$

In this section, we consider solving composite optimization problems

$$\min_{x \in \mathbb{R}^d} f(x) + \psi(x), \tag{2.72}$$

where $f(\cdot)$ is smooth convex but $\psi(\cdot)$ is possibly non-differentiable convex (e.g. $\|\cdot\|_1$). We want to show that the game analysis still applies to this problem. We just need to change the payoff function g to account for $\psi(x)$. Specifically, we consider the following two-players zero-sum game,

$$\min_{x} \max_{y} g(x,y) := \{\langle x, y\rangle - f^*(y) + \psi(x)\}. \tag{2.73}$$

Notice that the minimax value of the game is $\min_x f(x) + \psi(x)$, which is exactly the optimum value of the composite optimization problem. Let us denote the proximal operator as $\mathbf{prox}_{\lambda\psi}(v) = \operatorname{argmin}_x \left(\psi(x) + \frac{1}{2\lambda}\|x - v\|_2^2\right)$. [3] We have Algorithm 10. We remark that Algorithm 10 is essentially Algorithm 6, as the learners use the same stategies and the weighting scheme $\alpha_t = t$ is the same. The only difference is the new payoff function $g(x, y)$ (2.73).

In this new game, the x-player plays OMD$^+$ with the distance generating function $\phi_x = \frac{1}{2}\|x\|_2^2$, which leads to the following update,

$$\begin{aligned} x_t &= \operatorname*{argmin}_x \gamma(\alpha_t h_t(x)) + V_{x_{t-1}}(x) = \operatorname*{argmin}_x \gamma(\alpha_t\{\langle x, y_t\rangle + \psi(x)\}) + D^{\phi}_{x_{t-1}}(x) \\ &= \operatorname*{argmin}_x \phi(x) + \frac{1}{2\alpha_t\gamma}(\|x\|_2^2 + 2\langle \alpha_t\gamma y_t - x_{t-1}, x\rangle) = \mathbf{prox}_{\alpha_t\gamma\psi}(x_{t-1} - \alpha_t\gamma\nabla f(\widetilde{x}_t)). \end{aligned} \tag{2.74}$$

One can view Algorithm 10 as a variant of the so called "Accelerated Proximal Gradient"in Beck and Teboulle [25]. Yet, the design and analysis of our algorithm is simpler than that of Beck and Teboulle [25].

Theorem 13. *Denote $D := V_{x_0}(x^*)$. The weighted average of $\bar{x}_T$ in Algorithm 10 satisfies*

$$f(\bar{x}_T) - \min_x f(x) \leq O(\frac{LD}{T^2}).$$

Proof. Even though the payoff function $g(\cdot,\cdot)$ is a bit different, the proof still essentially follows the same line as Theorem 9 and Collorary 1, as y-player plays OPTIMISTICFTL and the x-player plays OMD$^+$.

[3] It is known that for some $\psi(\cdot)$, their corresponding proximal operations have closed-form solutions (see e.g. [220] for details).

$\square$

2.5.2.4 Related works

In recent years, there are growing interest in giving new interpretations of Nesterov's accelerated algorithms or proposing new varaints. For example, Tseng [259] gives a unified analysis for some Nesterov's accelerated algorithms Nesterov [206], Nesterov [207], Nesterov [208], using the standard techniques and analysis in optimization literature. Lessard, Recht, and Packard [168], Hu and Lessard [133] connects the design of accelerated algorithms with dynamical systems and control theory. Bubeck, Lee, and Singh [37] gives a geometric interpretation of the Nesterov's method for unconstrained optimization, inspired by the ellipsoid method. Flammarion and Bach [87] studies the Nesterov's methods and the HEAVYBALL method for quadratic non-strongly convex problems by analyzing the eigen-values of some linear dynamical systems. Allen-Zhu and Orecchia [11] proposes a variant of accelerated algorithms by mixing the updates of gradient descent and mirror descent and showing the updates are complementary. Diakonikolas1 and Orecchia [72],Diakonikolas and Orecchia [71] propose a primal-dual view that recovers several first-oder algorithms with careful discretizations of a continuous-time dynamic, which also leads to a new accelerated extra-gradient descent method. Cohen, Sidford, and Tian [57] show a simple acceleration proof of mirror prox operators and problems [215] and dual extrapolation Nesterov [210] based on solving the Fenchel game. Su, Boyd, and Candes [248], Wibisono, Wilson, and Jordan [274], Shi et al. [241] connect the acceleration algorithms with differential equations. Finally, we note an independent work Lan and Zhou [162], Lan [160] provide a game interpretation of Nesterov's accelerated method. In our work, we show a deeper connection with regret analysis in online learning and propose a modular framework that is not limited to Nesterov's method. We also note that in recent years there has emerged a lot of work where learning problems are treated as repeated games, and many researchers have been studying the relationship between game dynamics and provable convergence rates (see e.g. [4, 22, 113, 67, 194]).

2.5.3 Accelerated linear-rate method for strongly convex smooth problems

Algorithm 11 Accelerated Gradient with Linear Convergence

Given: L-smooth μ-strongly convex $f(\cdot)$, convex domain $\mathcal{K}$, arbitrary $w_0 \in \mathcal{K}$, iterations T, and a distance generating function $\phi(\cdot)$ that is 1-strongly convex, L_ϕ-smooth, and differentiable,

$$\begin{aligned}
\beta &\leftarrow \tfrac{1}{2}\sqrt{\tfrac{\mu}{L(1+L_\phi)}}, \gamma_t \leftarrow \alpha_t \\
z_t &\leftarrow (1-\beta)w_{t-1} + \beta v_{t-1} \\
v_t &\leftarrow \operatorname*{argmin}_{x\in\mathcal{K}} \sum_{s=1}^{t} \gamma_s \langle \nabla \tilde{f}(z_s), x\rangle + \mu\phi(x) \\
w_t &\leftarrow (1-\beta)w_{t-1} + \beta v_t
\end{aligned}$$

Iterative Description

$$\begin{aligned}
g(x,y) &:= \langle x, y\rangle - \tilde{f}^*(y) + \mu\phi(x) \\
&\quad \text{where } \tilde{f}(x) := f(x) - \mu\phi(x) \\
\alpha_1 &:= \tfrac{\mu}{2L(1+L_\phi)} \\
\alpha_t &:= \tfrac{\theta}{1-\theta}A_{t-1} \text{ for } t = 2, \dots, T, \\
\text{where } \theta &:= \tfrac{1}{2}\sqrt{\tfrac{\mu}{L(1+L_\phi)}} \\
\text{OAlg}^Y &:= \text{OPTIMISTICFTL}[\nabla f(v_0)] \\
\text{OAlg}^X &:= \text{FTRL}^+[\phi]
\end{aligned}$$

FGNRD Equivalence

Output: $w_T = \bar{x}_T$

Nesterov observed that, when $f(\cdot)$ is both μ-strongly convex and L-smooth, one can achieve a rate that is exponentially decaying in T (e.g. page 71-81 of [207]). It is natural to ask if the zero-sum game and regret analysis in the present work also recovers this faster rate in the same fashion. We answer this in the affirmative. Denote $\kappa := \frac{L}{\mu}$. In the following, we assume that the function $f(\cdot)$ is L-smooth with respect to some norm $\|\cdot\|$ and there exists a differentiable function $r(\cdot)$ that is L_ϕ-smooth and 1-strongly convex with respect to the same norm $\|\cdot\|$. Furthermore, assume $f(\cdot)$ is μ-strongly convex in the following sense (see also Section 3.3 of [160]),

$$f(z) \geq f(x) + \langle \nabla f(x), z - x\rangle + \mu D_x^\phi(z), \tag{2.75}$$

for all $z, x \in \mathcal{K}$, where $D_x^\phi(z)$ is the Bregman divegence. In the case that the norm is the l_2 norm, i.e. $\|\cdot\|_2$, we can define $\phi(x) := \frac{1}{2}\|x\|_2^2$ (and hence $L_\phi = 1$), and the strong convexity condition (2.75) becomes

$$f(z) \geq f(x) + \langle \nabla f(z), z - x \rangle + \frac{1}{2}\|z - x\|_2^2. \tag{2.76}$$

The function $\tilde{f}(x) := f(x) - \mu\phi(x)$ is a convex function for all $x \in \mathcal{K}$ (see e.g. [187]). Based on this property, we consider a new game

$$\tilde{g}(x, y) := \langle x, y \rangle - \tilde{f}^*(y) + \mu\phi(x), \tag{2.77}$$

where the minimax vale of the game is $V^* := \min_x \max_y \tilde{g}(x, y) = \min_x \tilde{f}(x) + \mu\phi(x) = \min_x f(x)$. In this game, the loss of the y-player in round t is $\alpha_t \ell_t(y) := \alpha_t(\tilde{f}^*(y) - \langle x_t, y \rangle)$, while the loss of the x-player in round t is a strongly convex function $\alpha_t h_t(y) := \alpha_t(\langle x, y_t \rangle + \mu\phi(x))$. We have the following theorem

Theorem 14. *Suppose that the function $f(\cdot)$ is L-smooth with respect to some norm $\|\cdot\|$ and $\phi(\cdot)$ is differentiable, L_ϕ-smooth, and 1-strongly convex with respect to the same norm. Assume that the function $f(\cdot)$ is μ-strongly convex in the sense of (2.75). Define the game $\tilde{g}(x, y) := \langle x, y \rangle - \tilde{f}^*(y) + \mu\phi(x)$. If the y-player plays* OptimisticFTL*: $y_t \leftarrow \nabla \tilde{f}(\tilde{x}_t)$ and the x-player plays* FTRL$^+$*: $x_t \leftarrow \arg\min_{x \in \mathcal{X}} \sum_{s=1}^{t} \alpha_s h_s(x) + R(x)$, where $R(x) := \alpha_0 \mu\phi(x)$, then the weighted average points $(\bar{x}_T, \bar{y}_T)$ would be an $O(\exp(-\frac{T}{2\sqrt{1+L_\phi}\sqrt{\kappa}}))$-approximate equilibrium of the game, where the weights $\alpha_0, \alpha_1, \ldots, \alpha_T$ satisfy $\frac{\alpha_1}{\alpha_0} \leq \frac{\mu}{2L(1+L_\phi)}$ and that for $t \geq 2$, $\frac{\alpha_t}{A_t} = \frac{1}{2\sqrt{1+L_\phi}}\sqrt{\frac{\mu}{L}}$. This implies that $f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) = O(\exp(-\frac{T}{2\sqrt{1+L_\phi}\sqrt{\kappa}}))$.*

Proof. As the proof of Lemma 13, we first bound the regret of the y-player as follows.

$$
\begin{array}{rcl}
\sum_{t=1}^{T} \alpha_t \ell_t(\widetilde{y}_t) - \alpha_t \ell_t(y^*) & \leq & \sum_{t=1}^{T} \alpha_t \langle x_{t-1} - x_t, \widetilde{y}_t - \hat{y}_{t+1} \rangle \\
\text{(Eqns. 2.56, 2.58)} & = & \sum_{t=1}^{T} \alpha_t \langle x_{t-1} - x_t, \nabla \tilde{f}(\widetilde{x}_t) - \nabla \tilde{f}(\bar{x}_t) \rangle \\
\text{(Hölder's Ineq.)} & \leq & \sum_{t=1}^{T} \alpha_t \| x_{t-1} - x_t \| \| \nabla \tilde{f}(\widetilde{x}_t) - \nabla \tilde{f}(\bar{x}_t) \|_* \\
 & = & \sum_{t=1}^{T} \alpha_t \| x_{t-1} - x_t \| \\
 & & \quad \times \| \nabla f(\widetilde{x}_t) - \mu \nabla \phi(\widetilde{x}_t) - \nabla f(\bar{x}_t) + \mu \nabla \phi(\bar{x}_t) \|_* \\
\text{(triangle inequality)} & \leq & \sum_{t=1}^{T} \alpha_t \| x_{t-1} - x_t \| \\
 & & \quad \times (\| \nabla f(\widetilde{x}_t) - \nabla f(\bar{x}_t) \|_* + \mu L_\phi \| \bar{x}_t - \widetilde{x}_t \|) \\
(L\text{-smoothness and } L \geq \mu) & \leq & L(1 + L_\phi) \sum_{t=1}^{T} \alpha_t \| x_{t-1} - x_t \| \| \widetilde{x}_t - \bar{x}_t \| \\
\text{(Eqn. 2.60)} & = & L(1 + L_\phi) \sum_{t=1}^{T} \frac{\alpha_t^2}{A_t} \| x_{t-1} - x_t \| \| x_{t-1} - x_t \|
\end{array}
$$

Therefore, the regret satisfies

$$
\boldsymbol{\alpha}\text{-}\mathrm{REG}^y \leq L(1 + L_\phi) \sum_{t=1}^{T} \frac{\alpha_t^2}{A_t} \| x_{t-1} - x_t \|^2. \tag{2.78}
$$

□

For the x-player, denote $\tilde{A}_t := \sum_{s=0}^{t} \alpha_s$. Notice that this is different from $A_t := \sum_{s=1}^{t} \alpha_s$. Then, according to Lemma 7, its regret is

$$
\boldsymbol{\alpha}\text{-}\mathrm{REG}^x \leq R(x^*) - R(x_0) - \sum_{t=1}^{T} \frac{\mu \tilde{A}_{t-1}}{2} \| x_{t-1} - x_t \|_2^2, \tag{2.79}
$$

where $x_0 = \arg\min_x R(x)$. Summing (2.78) and (2.79), we have

$$
\boldsymbol{\alpha}\text{-}\mathrm{REG}^y + \boldsymbol{\alpha}\text{-}\mathrm{REG}^x \leq R(x^*) - R(x_0) + \sum_{t=1}^{T} \left(\frac{L(1 + L_\phi) \alpha_t^2}{A_t} - \frac{\mu \tilde{A}_{t-1}}{2} \right) \| x_{t-1} - x_t \|_2^2. \tag{2.80}
$$

By choosing the weight $\{\alpha_t\}$ to satisfy $\frac{\alpha_1}{\alpha_0} \leq \frac{\mu}{2L(1+L_\phi)}$ and that for $t \geq 2$, $\frac{\alpha_t}{A_t} = \frac{1}{2}\sqrt{\frac{\mu}{L(1+L_\phi)}}$, the coefficient distance terms will be non-positive, i.e. $(\frac{L(1+L_\phi)\alpha_t^2}{A_t} - \frac{\mu\tilde{A}_{t-1}}{2}) \leq 0$, which means that the distance terms will cancel out. To see this, let $\frac{\alpha_t}{A_t} = \theta$ for some constant $\theta > 0$, we have that

$$\begin{aligned} \frac{L(1+L_\phi)\alpha_t^2}{A_t} - \frac{\mu\tilde{A}_{t-1}}{2} &= L(1+L_\phi)\theta^2 A_t - \frac{\mu}{2}(A_t(1-\theta)+\alpha_0) \\ &\leq A_t\left(L(1+L_\phi)\theta^2 - \frac{\mu}{2}(1-\theta)\right). \end{aligned} \tag{2.81}$$

So it suffices to have that $L(1+L_\phi)\theta^2 - \frac{\mu}{2}(1-\theta) \leq 0$, which can be guaranteed by choosing $\theta = \frac{1}{2}\sqrt{\frac{\mu}{L(1+L_\phi)}}$.

Therefore, the optimization error ϵ after T iterations satisfies that

$$\begin{aligned} \epsilon &\leq \frac{\boldsymbol{\alpha}\text{-}\textsc{Reg}^y + \boldsymbol{\alpha}\text{-}\textsc{Reg}^x}{A_T} \leq \frac{1}{A_1}\frac{A_1}{A_2}\cdots\frac{A_{T-1}}{A_T}(R(x^*)-R(x_0)) \\ &= \frac{1}{A_1}(1-\frac{\alpha_2}{A_2})\cdots(1-\frac{\alpha_T}{A_T})(R(x^*)-R(x_0)) \\ &\leq \frac{1}{A_1}(1-\frac{\alpha_2}{\tilde{A}_2})\cdots(1-\frac{\alpha_T}{\tilde{A}_T})(R(x^*)-R(x_0)) \\ &\leq (1-\frac{1}{2\sqrt{1+L_\phi}\sqrt{\kappa}})^{T-1}\frac{R(x^*)-R(x_0)}{A_1}, \end{aligned} \tag{2.82}$$

which is $O\left((1-\frac{1}{2\sqrt{1+L_\phi}\sqrt{\kappa}})^T\right) = O\left(\exp\left(-\frac{1}{2\sqrt{1+L_\phi}\sqrt{\kappa}}T\right)\right)$.

2.6 New algorithms

2.6.1 Boundary Frank-Wolfe

We observe that the meta-algorithm previously discussed assumed that the y-player (i.e. the player who plays gradients) was first to act, followed by the x-player who was allowed to be prescient. Here we reverse their roles, and we instead allow the y-player to be prescient. The new meta-algorithm is described in Algorithm 13. We are going to show that this framework lead to a new projection-free algorithm that works for non-smooth objective functions. Specifically, if the constraint set is strongly convex, then this exhibits a novel projection free algorithm that grants a $O(\log T/T)$ convergence even for non-smooth objective functions. The result relies on very recent work showing that FTL for strongly convex sets Huang et al. [137] grants a $O(\log T)$ regret rate. Prior work has considered strongly convex decision sets [100], yet with the additional assumption that the objective is smooth and strongly convex, leading to $O(1/T^2)$ convergence. *Boundary Frank-Wolfe* requires neither smoothness nor strongly convexity of the objective. What we have shown, essentially, is that a strongly convex boundary of the constraint set can be used in place of smoothness of $f(\cdot)$ in order to achieve $O(1/T)$ convergence.

Algorithm 12 Boundary Frank-Wolfe

1: **Input:** Init. $x_1 \in \mathcal{K}$.
2: **for** $t = 2, 3 \ldots, T$ **do**
3: $\quad x_t \leftarrow \text{argmin}_{x\in\mathcal{K}} \frac{1}{t-1}\sum_{s=1}^{t-1}\langle x, \partial f(x_s)\rangle$
4: **end for**
5: Output: $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$

Algorithm 13 Modified meta-algorithm, swapped roles

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad x_t := \text{OAlg}^X(g(\cdot, y_1), \ldots, g(\cdot, y_{t-1}))$
3: $\quad y_t := \text{OAlg}^Y(g(x_1, \cdot), \ldots, g(x_{t-1}, \cdot), g(x_t, \cdot))$
4: **end for**
5: Output: $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$ and $\bar{y}_T := \frac{1}{T}\sum_{t=1}^{T} y_t$

Theorem 15. *Algorithm 12 is a instance of Algorithm 13 if **(I)** Init. x_1 in Alg 12 equals x_1 in Alg. 13; **(II)** Alg. 13 sets OAlg^X $:=$ FTL; and **(III)** Alg. 13 sets OAlg^Y $:=$ BESTRESP^+. Furthermore, when the constraint set $\mathcal{X} \leftarrow \mathcal{K}$ is λ-strongly convex, and $\sum_{s=1}^{t}\partial f(x_s)$ has non-zero norm, then*

$$f(\bar{x}_T) - \min_{x\in\mathcal{K}} f(x) = O(\frac{M\log T}{\lambda L_T T})$$

where $M := \sup_{x\in\mathcal{K}}\|\partial f(x)\|$, $\Theta_t := \sum_{s=1}^{t}\frac{1}{t}\partial f(x_s)$, and $L_T := \min_{1\le t\le T}\|\Theta_t\|$.

Proof. Note that we have chosen the weighting scheme be $\alpha_t = 1$ for all t. Since y-player plays BESTRESP$^+$, its regret is 0. For the x-player, we use Lemma 3, its has regret which satisfies $\boldsymbol{\alpha}$-$\textsc{Reg}^x \leq O(\frac{M \log T}{\lambda L_T})$. So, by summing the average regrets of both players, we obtain the result, i.e.

$$f(\bar{x}_T) - \min_{x \in \mathcal{K}} f(x) \leq \tfrac{1}{T}(\boldsymbol{\alpha}\text{-}\textsc{Reg}^x[\text{FTL}] + \boldsymbol{\alpha}\text{-}\textsc{Reg}^y[\textsc{BestResp}^+]) \leq O(\tfrac{M \log T}{\lambda L_T T}). \tag{2.83}$$

□

Note that the rate depends crucially on L_T, which is the smallest averaged-gradient norm computed during the optimization. Now let us discuss when the boundary FW works; namely, the condition that causes the cumulative gradient being nonzero. If a linear combination of gradients is $\mathbf{0}$ then clearly $\mathbf{0}$ is in the convex hull of subgradients $\partial f(x)$ for boundary points x. Since the closure of $\{\nabla f(x) | x \in \mathcal{K}\}$ is convex, according to Theorem 4, this implies that $\mathbf{0}$ is in $\{\nabla f(x) | x \in \mathcal{K}\}$. If we know in advance that $\mathbf{0} \notin \text{closure}(\{\nabla f(x) | x \in \mathcal{K}\})$ we are assured that the cumulative gradient will not be $\mathbf{0}$. Hence, the proposed algorithm may only be useful when it is known, a priori, that the solution x^* will occur not in the interior but on the boundary of $\mathcal{K}$. It is indeed an odd condition, but it does hold in many typical scenarios. One may add a perturbed vector to the gradient and show that with high probability, L_T is a non-zero number. The downside of this approach is that it would generally grant a slower convergence rate; it cannot achieve $\log(T)/T$ as the inclusion of the perturbation requires managing an additional trade-off.

2.6.2 Gauge Frank-Wolfe

We propose a new FRANK-WOLFE like algorithm that not only requires a linear oracle but also enjoys $O(1/T^2)$ rate on all the strongly convex constraint sets that contain the origin, like l_p ball and Schatten p ball with $p \in (1, 2]$. To describe our algorithm, denote $\mathcal{K}$ be any closed convex set that contains the origin. Define "gauge function" of $\mathcal{K}$ [92, 195] as

$$\gamma_{\mathcal{K}}(x) := \inf\{c \geq 0 : \frac{x}{c} \in \mathcal{K}\}. \tag{2.84}$$

Notice that, for a closed convex $\mathcal{K}$ that contains the origin, $\mathcal{K} = \{x \in \mathbb{R}^d : \gamma_{\mathcal{K}}(x) \leq 1\}$. Furthermore, the boundary points on $\mathcal{K}$ satisfy $\gamma_{\mathcal{K}}(x) = 1$.

Next we provide a characterization of sets based on their gauge function.

Definition 1 (λ-Gauge set)**.** *Let $\mathcal{K}$ be a closed convex set which contains the origin. We say that $\mathcal{K}$ is λ-Gauge if its squared gauge function, $\gamma^2_{\mathcal{K}}(\cdot)$, is λ-strongly-convex.*

This property captures a wide class of constraints. Among these are l_p balls, Schatten p balls, and the Group (s, p) ball [100]. In fact, Theorem 4 in [222] and Theorem 2 in [198] show that for any centrally symmetric strongly convex set $\mathcal{K}$ that contains the origin, the gauge function $\gamma^2_{\mathcal{K}}(\cdot)$ is strongly convex w.r.t. the induced gauge norm $\gamma_{\mathcal{K}}(\cdot)$ on $\mathcal{K}$.

We introduce a family of FTRL$^+$ algorithms that rely solely on a linear oracle, and we believe this is a novel approach to online linear optimization problems. The restriction we require is that the regularizer $R(\cdot)$ is chosen as the *squared gauge function* $\gamma^2_{\mathcal{K}}(\cdot)$ for the decision set $\mathcal{K}$ of the learner. Here we will assume[4] for every t that $\ell_t(\cdot) = \langle l_t, \cdot\rangle$ for some vector l_t, hence FTRL$^+$ (2.24) reduces to

$$x_t = \operatorname*{argmin}_{x\in\mathcal{K}} \eta\langle L_t, x\rangle + \gamma^2_{\mathcal{K}}(x), \tag{2.85}$$

where $L_t = l_1 + \ldots + l_t$. Denote bndry($\mathcal{K}$) as the boundary of the constraint set $\mathcal{K}$. We can reparameterize the above optimization, by observing that any point $x \in \mathcal{K}$ can be written as ρz where $z \in$ bndry($\mathcal{K}$), and $\rho \in [0, 1]$. Hence we have

$$\min_{\rho\in[0,1]} \min_{z\in\text{bndry}(\mathcal{K})} \eta\langle L_t, \rho z\rangle + \gamma^2_{\mathcal{K}}(\rho z) \;=\; \min_{\rho\in[0,1]} \left(\min_{z\in\text{bndry}(\mathcal{K})} \eta\langle L_t, z\rangle\right)\rho + \rho^2. \tag{2.86}$$

[4] One can reduce any arbitrary convex loss to the linear loss case by convexity $\ell_t(x) - \ell(x^*) \leq \langle \partial f_t(x), x - x^*\rangle$ ([236, 225]).

Algorithm 14 Gauge Frank-Wolfe (smooth convex $f(\cdot)$)

1: Let $\{\alpha_t = t\}$ be a T-length weight sequence.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: The y-player plays OPTIMISTICFTL: $y_t = \nabla f(\tilde{x}_t)$.
4: The x-player plays BTRL:
5: Compute $(\hat{x}_t, \rho_t) = \underset{x \in \mathcal{K}, \rho \in [0,1]}{\arg\min} \sum_{s=1}^{t} \rho \langle x, \alpha_s y_s \rangle + \frac{1}{\eta}\rho^2$ and set $x_t = \rho_t \hat{x}_t$.
6: **end for**
7: Output $\bar{x}_T := \frac{\sum_{s=1}^{T} \alpha_s x_s}{\sum_{s=1}^{T} \alpha_t}$.

We are able to remove the dependence on the gauge function since it is homogeneous, $\gamma_{\mathcal{K}}(\rho x) = |\rho| \gamma_{\mathcal{K}}(x)$, and is identically 1 on the boundary of $\mathcal{K}$. The inner minimization reduces to the linear optimization $z^* := \mathrm{argmin}_{z \in \mathcal{K}} \langle L_t, z \rangle$, and the optimal ρ is

$$\rho = \max(0, \min(1, -(\eta/2)\langle L_t, z^* \rangle)). \tag{2.87}$$

Theorem 16. *Suppose the constraint set $\mathcal{K}$ is a λ-Gauge set. Assume that the function $f(\cdot)$ is L-smooth convex with respect to the induced gauge norm $\gamma_{\mathcal{K}}(\hat{x})$. Suppose that the step size η satisfies $\frac{1}{CL} \leq \frac{\eta}{\lambda} \leq \frac{1}{4L}$ for some constant $C \geq 4$. Then, the output $\bar{x}_T$ of Algorithm 14 satisfies*

$$f(\bar{x}_T) - \min_{x \in \mathcal{K}} f(x) \leq \frac{2CL\gamma_{\mathcal{K}}^2(x^*)}{\lambda T^2},$$

Proof. We have just shown that line 4-5 is due to that the x-player plays FTRL$^+$ with the squared of the guage function as the regularizer. So Algorithm 14 is an instance of the meta-algorithm, and we can invoke Corollary 1 to obtain the convergence rate.

□

We want to emphasize again that our analysis does not need the function $f(\cdot)$ to be strongly convex to show $O(1/T^2)$ rate. On the other hand, Garber and Hazan [100] shows the $O(1/T^2)$ rate under the additional assumption that the function is strongly convex.

2.6.3 A Fast Parallelizable Projection-Free Algorithm for the Nuclear-Norm-Ball Constraint

In this section, we consider smooth convex optimization with a bounded nuclear norm constraint,

$$\min_{W \in \mathcal{NB}_{d_1,d_2}(r)} f(W), \tag{2.88}$$

where $\mathcal{NB}_{d_1,d_2}(r)$ denotes the nuclear norm ball in $\mathbb{R}^{d_1 \times d_2}$ with radius r, defined as

$$\mathcal{NB}_{d_1,d_2}(r) := \left\{ W \in \mathbb{R}^{d_1 \times d_2} : \sum_{i=1}^{d_1 \wedge d_2} \sigma_i(W) \leq r \right\} \tag{2.89}$$

where $\wedge$ is the $\min$ operator and $\sigma_i(W)$ denotes the i^{th} singular value of W. This optimization problem is an important task for many applications in machine learning and signal processing, including matrix completion and collaborative filtering (e.g. [43, 239, 245, 123]), phase retrieval (e.g. [41]), affine rank minimization problems (e.g. [141, 228]), robust PCA (e.g. [42]), multi-task learning (e.g [142]), multi-class classification (e.g. [79, 293]), distance metric learning (e.g. [281, 286]), kernel matrix learning (e.g. [118]), learning polynomial networks (e.g. [182]), and more. The typically large dimensions d_1, d_2 that arise in common ML tasks has led to great interest in the development of a highly efficient method to solve (2.88).

A natural approach to solve (2.88) is projected gradient descent (PGD), where one performs alternating gradient updates followed by nuclear norm projections. But this last step, the projection, requires an expensive singular value decomposition (SVD) on each iteration whose complexity scales as $O(d_1 d_2 (d_1 \wedge d_2))$, cubic in the dimension (see e.g. [126]). On the other hand, the Frank-Wolfe method (Frank and Wolfe [91]) has a key benefit for dealing with the ball constraint: each iteration involves solving a linear optimization oracle (LMO) of the form $\arg\max_{x \in \mathcal{X}} \langle x, v \rangle$. When the constraint is the nuclear norm ball, i.e. $\mathcal{X} = \mathcal{NB}_{d_1,d_2}(r)$, the linear optimization problem reduces to computing the leading singular vector of (the negative of) the gradient matrix $-\nabla f(W)$ (see e.g. [126, 139]). In

Algorithm	Convergence rate	# Computations in iteration t
Projected Gradient Descent	$O\left(\frac{Lr^2}{T}\right)$	c-SVD
Accelerated PGD	$O\left(\frac{Lr^2}{T^2}\right)$	c-SVD
Frank-Wolfe	$O\left(\frac{Lr^2}{T}\right)$	c-LMO
this work	$\tilde{O}\left(\max\left\{\frac{Lr\log(d_1+d_2)}{T^2}, \frac{G}{T}\right\}\right)$	c-MEV $\times\, t$

Table 2.3: Comparison of first-order methods for solving Problem 2.88. The second column (convergence rate) is the optimization error $f(W_T) - \min_{W \in \mathcal{NB}_{d_1,d_2}(r)} f(W)$ in iteration t. Here r is the radius of the ball, L is the smoothness constant of $f(\cdot)$, and G is the constant that bounds the spectral norm of the gradient (i.e. $\|\nabla f(\cdot)\|_2 \leq G$ for all $W \in \mathcal{NB}_{d_1,d_2}(r)$). The last column is the cost per iteration, where c-SVD represents the cost of approximating a singular value decomposition (SVD), which is $O(d_1 d_2 (d_1 \wedge d_2))$ in practice; c-LMO is the cost of approximately solving the linear optimization oracle of Frank-Wolfe, which is $O(d_1 d_2)$ in practice; c-MEV is the cost of computing a matrix exponential-vector product, which is $O(d_1 d_2)$ in practice. The convergence rate of our method is better than the baselines under some reasonable conditions, e.g. when the largest gradient norm satisfies $G \leq Lr^2$ or when the radius r is large. Please see the main text (Section 2.6.3.3) for the discussion.

practice, one can approximate the top singular vector efficiently via standard approaches like power iteration or the Lanzcos algorithm, where the complexity is roughly proportional to the number of non-zeros in the input matrix (see e.g. [290, 59]). The cost of approximately computing the singular vector is $O(d_1 d_2)$ in the worst case, up to log factors (see e.g. [117]). Therefore, it is observed that the Frank-Wolfe method significantly improves performance compared to PGD or Accelerated PGD [206, 208] due to its cheap iteration cost (see e.g. [97, 126]).

A major concern of the Frank-Wolfe method is that it has a suboptimal $O(Lr^2/T)$ convergence rate for the smooth convex problems, compared to $O(Lr^2/T^2)$ of the accelerated gradient methods (see e.g. [211]), where L is the smoothness constant. To deal with this issue, some works have developed Frank-Wolfe-like algorithms that enjoy a better convergence rate. However, these results only apply to the case when the constraint set is a certain convex polytope [24, 33, 101, 103, 157, 32], or a strongly convex set [2, 100] and do not apply to the nuclear norm constraint studied in this work. On the other hand, Lan

[159] and Lan [160] developed an $O(1/T)$ lower bound for any algorithm that relies on the linear optimization oracle to generate the iterates for a smooth convex problem on the simplex, which might imply the hardness to get a rate beyond $O(1/T)$ for the nuclear norm ball constraint without additional assumptions. In this work, we propose a projection-free algorithm that enjoys a provably better convergence rate than the $O(Lr^2/T)$ rate of Frank-Wolfe and Projected Gradient Descent (PGD) under some reasonable conditions. As can be seen from Table 4.1, our algorithm has an advantage when the radius r is large. This improvement helps especially for some applications like matrix completion (e.g. Jaggi and Sulovský [140]). We will return to this point in details in the later sections.

Our algorithm has an additional advantage: it naturally lends itself to a simple parallelization scheme. In each iteration t, our algorithm must compute an average $\frac{1}{m_t}\sum_{i=1}^{m_t}\Psi_{u_i}(\cdot)$, where $m_t \approx t$ and where $\Psi_{u.}(\cdot)$ is a special oracle that maps a symmetric matrix to the spectrahedron in a randomized fashion that requires a matrix exponential vector product. This can be efficiently approximated by the Lanczos method, with complexity in the same ballpark as the Frank-Wolfe LMO which requires matrix-vector products and costs $O(d_1 d_2)$. Furthermore, by exploiting multiple processing units that are pervasively available in modern machines, the average $\frac{1}{m_t}\sum_{i=1}^{m_t}\Psi_{u_i}(\cdot)$ is embarrassingly parallelizable as each term can be independently and simultaneously computed.

2.6.3.1 Preliminaries

Smooth convex function We assume that the optimization problem is L-smooth convex w.r.t. the nuclear norm $\|\cdot\|$. This means that $f(\cdot)$ is differentiable everywhere (Vandenberghe [262]) and that it has Lipschitz continuous gradient $\|\nabla f(W) - \nabla f(Z)\|_* \leq L\|W - Z\|$, where $\|\cdot\|_*$ denotes the dual norm which is the spectral norm $\|\cdot\|_2$ (Tibshirani [255]).

Spectrahedron and an associated operator $\Psi_u(\cdot)$ We denote $\mathcal{S}_d$ the set of symmetric $d \times d$ matrices and denote the *spectrahedron* as

$$\Delta_d := \{X \in \mathcal{S}_d \;:\; X \succeq 0, \mathrm{Tr}(X) = 1\}, \tag{2.90}$$

which is the space of $d \times d$ real positive semi-definite symmetric matrices whose trace equals 1. For a symmetric matrix $X \in \mathcal{S}_d$ with $d := d_1 + d_2$, we will use the following factorization of X,

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} \\ X^{(2)\top} & X^{(3)} \end{bmatrix}, \tag{2.91}$$

where the dimension of the sub-matrix $X^{(1)}$ is $d_1 \times d_1$, $X^{(2)}$ is $d_1 \times d_2$, and $X^{(3)}$ is $d_2 \times d_2$.

We will need the operator $\Psi_u(\cdot) : \mathcal{S}_d \to \Delta_d$, defined as

$$\Psi_u(D) := \frac{\exp(D/2)uu^\top \exp(D/2)}{u^\top \exp(D)u} = \frac{v_u v_u^\top}{\|v_u\|^2}, \tag{2.92}$$

where D is a symmetric matrix in $\mathcal{S}_d$, $\exp(D) = \sum_{k=0}^{\infty} \frac{1}{k!} D^k$ is its matrix exponential, and $v_u := \exp(D/2)u$ with $u \sim \text{Uni}(\mathbb{S}_d)$, i.e. uniformly sampled from the unit sphere $\mathbb{S}_d$. From (3.92), we see that the computation of $\Psi_u(D)$ needs a matrix exponential-vector product, which can be done by Lanczos method , see e.g. the discussion in Section 3 of Carmon et al. [45]. We also denote $\bar{\Psi}(\cdot) := \mathbb{E}_u[\Psi_u(\cdot)] : \mathbb{S}_d \to \Delta_d$ as follows,

$$\bar{\Psi}(D) = \mathbb{E}_u\left[\frac{\exp(D/2)uu^\top \exp(D/2)}{u^\top \exp(D)u}\right] = \mathbb{E}_u\left[\frac{v_u v_u^\top}{\|v_u\|^2}\right], \tag{2.93}$$

where the expectation is over the random draw from the unit sphere $u \sim \text{Uni}(\mathbb{S}_d)$. The function $\bar{\Psi}(D)$ is a gradient of the function $\bar{\psi}(D) := \mathbb{E}_u[\log(u^\top \exp(D)u)]$, which is a continuously twice differentiable, convex spectral function (Lewis and Sendov [172]).

The spectrahedron operator $\bar{\Psi}(\cdot)$ is the key to a recent breakthrough by Carmon et al. [45] in *online learning* literature for a randomized sketch of the celebrated Matrix Multiplicative Weight (MMW) algorithm (Tsuda, Rätsch, and Warmuth [260], Warmuth and Kuzmin [272], Arora, Hazan, and Kale [18]). In the online setting, in each round the online learner plays an action in the spectrahedron $X_t \in \Delta_d$, and the adversary supplies a symmetric matrix $L_t \in \mathcal{S}_d$, and the player suffers a loss $\langle X_t, L_t \rangle := \mathrm{tr}(L_t X_t)$. The goal of the online player is to minimize the regret, $\text{Regret}_T := \sum_{t=1}^{T} \langle L_t, X_t \rangle - \inf_{X \in \Delta_d} \sum_{t=1}^{T} \langle L_t, X \rangle$, where the second term is the loss of the best single action in hindsight. MMW is a celebrated algorithm for this setting, and it has wide applications in machine learning and theoretical computer science. The update is $X_t = \frac{\exp(-\eta L_{1:t-1})}{\mathrm{tr}\left(\exp(-\eta L_{1:t-1})\right)}$, where $L_{1:t-1} := \sum_{s=1}^{t-1} L_s$. The update in general is expensive because computing matrix exponential needs an eigendecomposition whose cost is proportional to the cubic size of the matrix in practice, i.e. $O(d^3)$. On the other hand, Carmon et al. [45] propose a simple randomized algorithm with a cheaper computational cost that enjoys an $O(\sqrt{T})$ *expected* regret guarantee as MMW. The algorithm efficiently updates the action in each round according to $X_t = \Psi_{u_t}(-\eta L_{1:t-1})$, which only needs a matrix exponential-vector product and can be efficiently approximated by Lanczos method that costs only $O(d^2)$ in practice. Our algorithm adopts this random projection oracle $\Phi_u(\cdot)$. While it is possible to obtain an $O(\frac{1}{\sqrt{T}})$ convergence rate in expectation by applying the standard online-to-batch conversion (e.g. [235]) to the algorithm of Carmon et al. [45], we develop a new algorithm that has a faster convergence rate in this work.

2.6.3.2 Equivalent optimization problem

We will consider an equivalent optimization problem of (2.88), which is optimizing over the spectrahedron,

$$\min_{X \in \Delta_{d_1+d_2}} F_r(X) := f(2rX^{(2)}). \tag{2.94}$$

Lemma 14. *(Lemma 1 in Garber [97] and Lemma 1 in Jaggi and Sulovský [140]) Consider* $\min_{X \in \Delta_{d_1+d_2}} F_r(X) := f(2rX^{(2)})$. *Suppose that* $X \in \Delta_{d_1+d_2}$ *satisfies* $F_r(X) - F_r(X^*) \leq \epsilon$, *where $X^* \in \Delta_{d_1+d_2}$ is a minimizer of $F_r(\cdot)$ over $\Delta_{d_1+d_2}$. Then,*

$$f(2rX^{(2)}) - \min_{X \in \mathcal{NB}_{d_1,d_2}(r)} f(X) \leq \epsilon \tag{2.95}$$

and that $2rX^{(2)} \in \mathcal{NB}_{d_1,d_2}(r)$.

Remark 1: Lemma 14 shows the equivalency between problem (2.88) and (2.94). Solving (2.94) over the spectrahedron $\Delta_{d_1+d_2}$ is equivalent to solving problem (2.88). Note that for $X \in \Delta_{d_1+d_2}$ the gradient $\nabla F_r(X) \in \mathcal{S}_{d_1+d_2}$ is given by

$$\nabla F_r(X) = \begin{bmatrix} 0_{d_1 \times d_1} & \nabla f(2rX^{(2)}) \\ \nabla f(2rX^{(2)})^\top & 0_{d_2 \times d_2} \end{bmatrix}. \tag{2.96}$$

In the following, we will denote the dimension $d := d_1 + d_2$.

Remark 2: We will assume that $F_r(\cdot)$ is $\hat{L}$-smooth w.r.t. the nuclear norm $\|\cdot\|$ over Δ_d. Suppose that the original function $f(\cdot)$ is L-smooth over the nuclear norm ball $\mathcal{NB}_{d_1,d_2}(r)$. Let us discuss the relation between the smoothness constant $\hat{L}$ of $F_r(\cdot)$ and L of $f(\cdot)$. Denote $M_1 := \begin{bmatrix} I_{d_1}, 0_{d_1 \times d_2} \end{bmatrix}$ and $M_2 = \begin{bmatrix} 0_{d_1 \times d_2} \\ I_{d_2} \end{bmatrix}$. We have that

$$\begin{aligned} &\|\nabla F_r(X) - \nabla F_r(Z)\|_* = \|\nabla f(2rX^{(2)}) - \nabla f(2rZ^{(2)})\|_* \\ &\leq 2rL\|X^{(2)} - Z^{(2)}\| = 2rL\|M_1 X M_2 - M_1 Z M_2\| \\ &\leq 2rL\|X - Z\|, \end{aligned} \tag{2.97}$$

where we use the fact that the nuclear norm of a matrix product $\|AB\|$ satisfies $\|AB\| \leq \sigma_{\max}(A)\|B\|$, see e.g. Huang, Chen, and Hu [136], and that the largest singular values $\sigma_{\max}(M_1) = \sigma_{\max}(M_2) = 1$. So we see that $\hat{L} \leq 2rL$.

Algorithm 15 Proposed algorithm for solving (2.88)

1: Set parameter $\delta > 0$, $\beta_t = \frac{2}{t+1}$, $\eta \leq \frac{1}{36\hat{L}}$, and $m_t = \max\{\lceil \log(4d/\delta) \rceil, t\}$.
2: Init: $W_0 = X_0 \in \Delta_d$ and $G_0 = 0_{d\times d}$ with $d = d_1 + d_2$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: $\quad Z_t = (1-\beta_t)W_{t-1} + \beta_t X_{t-1}$.
5: $\quad G_t = G_{t-1} - \eta t \nabla F_r(Z_t) = G_{t-1} - \eta t \times \begin{bmatrix} 0_{d_1\times d_1} & \nabla f(2rZ_t^{(2)}) \\ \nabla f(2rZ_t^{(2)})^\top & 0_{d_2\times d_2} \end{bmatrix}$.
6: $\quad X_t = \frac{1}{m_t}\sum_{j_t=1}^{m_t} \Psi_{u_{j_t}}(G_t)$, where each $u_{j_t} \sim \mathrm{Uni}(\mathbb{S}^d)$. ***easily done in parallel***
7: $\quad W_t = (1-\beta_t)W_{t-1} + \beta_t X_t$.
8: **end for**
9: Output $2rW_T^{(2)} \in \mathcal{NB}_{d_1,d_2}(r)$.

2.6.3.3 Main result

Algorithm 15 shows the proposed algorithm. Similar to those accelerated gradient methods (e.g. Nesterov, Lan and Zhou [208, 162]), it maintains two interleaving sequences $\{W_t\}$ and $\{Z_t\}$ such that the gradient is computed at an auxiliary variable Z_t instead of the primary variable W_t (line 5). Furthermore, like the Frank-Wolfe method, it has the steps of the "convex averaging" so that the iterate W_t and Z_t are always in the constraint set (line 4 and 7), as the outputs from the oracle (line 6) and the initial points (line 2) are all in the constraint set. We remark that line 6 is where the parallelization can be exploited for the oracle calls.

2.6.3.4 Convergence rate

Theorem 17. *Suppose that the function $F_r(\cdot)$ on (2.94) is $\hat{L}$-smooth with respect to the nuclear norm over the spectrahedron Δ_d and that the gradient norm of $f(\cdot)$ satisfies $\|\nabla f(\cdot)\|_2 \leq G$ over the ball $\mathcal{NB}_{d_1,d_2}(r)$. If $\eta \leq \frac{1}{36\hat{L}}$ and $\forall t, m_t \geq \log(4d/\delta)$, then with a constant probability $1-\delta$, the output $2rW_T^{(2)} \in \mathcal{NB}_{d_1,d_2}(r)$ of Algorithm 15 satisfies*

$$f(2rW_T^{(2)}) - \min_{X\in\mathcal{NB}_{d_1,d_2}(r)} f(X) \leq O\left(\frac{\hat{L}\log(d)}{T^2}\right) + \frac{c_1}{T^2}\sum_{t=1}^{T}\frac{1}{m_t} + \frac{c_2}{T^2}\sqrt{\sum_{t=1}^{T}\frac{t^2}{m_t}}, \quad (2.98)$$

where $c_1 := 192\hat{L}\log\frac{4d}{\delta}$ *and* $c_2 := \sqrt{448G^2\log\frac{2}{\delta}}$.

In Section 2.6.3.7 and Section 4.4.6 we will provide the analysis and the proof. We note that the convergence rate in Theorem 17 has two components — one is a fast rate term $O(\frac{\hat{L}\log d}{T^2}) = O(\frac{Lr\log d}{T^2})$, while the other is controlled by the number of oracle calls m_t.

Corollary 2. *Under the same setup as Theorem 17, if the number of oracle calls* m_t *to construct* x_t *is* $m_t = \max\{\lceil\log(4d/\delta)\rceil, t\}$*, then with a constant probability* $1-\delta$*, the output* $2rW_T^{(2)} \in \mathcal{NB}_{d_1,d_2}(r)$ *of Algorithm 15 satisfies*

$$f(2rW_T^{(2)}) - \min_{X\in\mathcal{NB}_{d_1,d_2}(r)} f(X) \leq O\left(\frac{\hat{L}\log(dT/\delta)}{T^2} + \frac{G\sqrt{\log(1/\delta)}}{T}\right). \tag{2.99}$$

2.6.3.5 Comparison of the convergence rates

Let us compare the convergence rates in the literature. The convergence rate of Frank-Wolfe for general smooth convex problem with a convex constraint set $\mathcal{X}$ is

$$f(X_T) - \min_{X\in\mathcal{X}} f(X) \leq \frac{2C_f}{T}, \tag{2.100}$$

where C_f is defined as $C_f := \sup_{\Omega}\frac{1}{\theta^2}\big(f(Z') - f(Z) + \langle Z'-Z, \nabla f(Z)\rangle\big)$ with the constraint $\Omega := \{Z, V \in \mathcal{X}, \theta \in R, Z' = Z + \theta(V-Z)\}$ (see e.g. Clarkson, Jaggi and Sulovský [56, 140]). The constant C_f can be upper-bounded by

$$C_f \leq \sup_{Z,V\in\mathcal{X}} L\|Z-V\|^2, \tag{2.101}$$

since smoothness implies that $f(Z') - f(Z) + \langle Z'-Z, \nabla f(Z)\rangle \leq L\|Z'-Z\|$ (Vandenberghe [262]). On the other hand, projected gradient descent (PGD) is known to have:

$$f(X_T) - \min_{X\in\mathcal{X}} f(X) \leq \frac{2L\|X_0 - X_*\|^2}{T}, \tag{2.102}$$

where X_* is one of the minimizers of $f(X)$ (see e.g. Section 3.2 of [38]). Furthermore, Accelerated PGD (e.g. Nesterov's method Nesterov, Tseng [211, 259]) has convergence rate,

$$f(X_T) - \min_{X \in \mathcal{X}} f(X) \leq \frac{2L\|X_0 - X_*\|^2}{T^2}. \tag{2.103}$$

Now let us identify conditions such that the rate of Algorithm 15 stated in Corollary 2 is better than the baselines. We consider two cases, which are *(A):* $O(\frac{\hat{L}\log(dT/\delta)}{T^2})$ being the dominant term (i.e. slower one) of the convergence rate (2.99) and *(B):* $O(\frac{G\sqrt{\log(1/\delta)}}{T})$ being the dominant term of the rate (2.99). The first case happens when $\hat{L}$ is large, i.e. $\hat{L} \gg G$, and T is not too large. In this case, our algorithm has a fast rate of $O(1/T^2)$ which matches that of Accelerated PGD, while Frank-Wolfe and PGD have the slow rate $O(1/T)$.

Moreover, the convergence rate of Algorithm 15 is better when r is large. Specifically, for a fixed T and δ, the constant factor of the convergence rate of Algorithm 15, i.e. $\hat{L}\log(dT/\delta) \leq 2Lr\log(dT/\delta)$, can be smaller than $4Lr^2$ of Frank-Wolfe (2.101), of PGD (2.102), and of Accelerated PGD (2.103). This happens when the radius r is large. For example, in the experiments of Jaggi and Sulovský [140], to obtain a good testing performance for solving the matrix completion problem, the authors set the radius of the nuclear norm ball to be $r = 4988$ for the MovieLens 100k dataset (a 943 by 1682 user-rating matrix) and $r = 18080$ for the MovieLens 1m dataset (a 6040 by 3706 user-rating matrix). On the other hand, the logarithm of dimension $d = d_1 + d_2$ is $\log d = \log(943 + 1682) = 7.87$ for MovieLens 100k and $\log d = 9.18$ for MovieLens 1m. Therefore, the constants $2Lr\log(dT/\delta)$ and $4Lr^2$ are in significantly different scales, which suggests that our algorithm can have a better performance over the baselines.

Now let us switch to the case when $O(\frac{G\sqrt{\log(1/\delta)}}{T})$ is the dominant term. In this case, Algorithm 15, Frank-Wolfe, and PGD all have the same $O(1/T)$ rate. However, they depend on different constants. Consider a point $\hat{W}$ in the ball $\mathcal{NB}_{d_1,d_2}(r)$ whose gradient norm is the *smallest* one among the points in the ball. Then, $G := \max_{W \in \mathcal{NB}_{d_1,d_2}(r)} \|\nabla f(W)\|_2 = \max_{W \in \mathcal{NB}_{d_1,d_2}(r)} \|\nabla f(W) - \nabla f(\hat{W}) + \nabla f(\hat{W})\|_2 \leq L\|W - \hat{W}\|_2 + \|\nabla f(\hat{W})\|_2 \leq$

$2Lr + \|\nabla f(\hat{W})\|_2$. Hence, if the smallest gradient norm satisfies $\|\nabla f(\hat{W})\|_2 \leq 2rL$, then $G \leq 4Lr$ is smaller than $4Lr^2$ of Frank-Wolfe and PGD when $r > 1$.

2.6.3.6 *Analysis of the computational cost*

In this subsection, we analyze the computational cost required for Algorithm 15 to reach at a point whose function value is ϵ-close to the optimal value of (2.88) and compare it with Frank-Wolfe. Corollary 1 states that Algorithm 15 needs $T = \tilde{O}(\max\{\sqrt{\frac{\hat{L}\log d}{\epsilon}}, \frac{G}{\epsilon}\})$ iterations to reach $f(2rW_T^{(2)}) - \min_{X \in \mathcal{NB}_{d_1,d_2}(r)} f(X) \leq \epsilon$. As Algorithm 15 describes, it needs computing $\frac{1}{m_t}\sum_{j_t=1}^{m_t} \Psi_{u_{j_t}}(G_t)$ in each iteration t. Each call to the oracle $\Psi_{u.}(\cdot)$ requires a matrix exponential-vector product (recall the definition in Section 3.2). The matrix exponential-vector product can be efficiently approximated by Lanczos method in $O(d_1 d_2)$ time, see e.g. the discussion in Section 3 of Carmon et al. [45] or Musco, Musco, and Sidford [201]. In our algorithm, the number of calls to the oracle $\Psi_{u.}(\cdot)$ in each iteration grows linearly with iteration t. The total number of oracle calls, and hence the total number of matrix exponential-vector products during the execution of the algorithm $\sum_{t=1}^{T} m_t$ is

$$\begin{aligned} &\sum_{t=1}^{T} \max\{\lceil \log(\frac{4d}{\delta})\rceil, t\} = \lceil \log(\frac{4d}{\delta})\rceil T + \frac{T(T+1)}{2} \\ &= O\left(\max\left\{\frac{\hat{L}\log(d)}{\epsilon}, \frac{G^2}{\epsilon^2}\right\}\right), \end{aligned} \tag{2.104}$$

if $\log(4d/\delta) \leq T$. Now let us compare this number with that of the Frank-Wolfe method. The Frank-Wolfe method needs $T = O(\frac{Lr^2}{\epsilon})$ iterations to achieve an ϵ error. In each iteration, it makes one linear optimization oracle call for computing the top singular vector of a gradient matrix (Hazan [126]). Therefore, the total number of oracle calls and hence the total number of top singular vector computations by Frank-Wolfe is $O(\frac{Lr^2}{\epsilon})$. The top singular vector can be efficiently approximated by power iteration or by the Lanczos method, and the cost is in the order of $O(d_1 d_2)$. So the cost of a single call to our oracle and the cost of a single call of that of Frank-Wolfe is similar. Algorithm 15 makes fewer number of oracle calls than Frank-Wolfe if $O(\max\{\frac{\hat{L}\log(d)}{\epsilon}, \frac{G^2}{\epsilon^2}\}) < O(\frac{Lr^2}{\epsilon})$, which holds when r is large as discussed. On the other hand, even if Algorithm 15 needs more number of oracle calls, its actual running time can be better than that of Frank-Wolfe due to the fact that calls to its oracle are embarrassingly easy to be parallelized.

Modern computational resources have multi-cores or multiple processing units, which enables conducting a task in a parallel fashion. Our algorithm can immediately benefit from parallel computing. Observe that parallelizing Step 6 of Algorithm 15, $X_t = \frac{1}{m}\sum_{j_t=1}^{m} \Psi_{u_{j_t}}(G_t)$, is embarrassingly easy — simply let each worker of the machine independently and simultaneously compute some $\Psi_{u_{j_t}}(G_t)$ in parallel. As the result, the actual time spent in computing X_t can be significantly reduced by the parallel computing. Since the actual running time is the number of iterations times the cost (computational time) per iteration, mathematically speaking, the actual running time is,

$$O(\max\{\frac{\hat{L}\log(d_1+d_2)}{\epsilon}, \frac{G^2}{\epsilon^2}\}) \times \frac{O(d_1 d_2)}{M}, \tag{2.105}$$

where $M \geq 1$ represents a factor of reduction due to the parallelization of the calls and could be viewed as the "effective" number of workers in a machine. Hence, the effective computational time (2.105) of Algorithm 15 can be better than Frank-Wolfe, which is $O(\frac{Lr^2}{\epsilon}) \times$ c-LMO $= O(\frac{Lr^2}{\epsilon}) \times O(d_1 d_2)$. On the other hand, it is not clear if parallelizing the calls to the linear optimization oracle of Frank-Wolfe is feasible and we are not aware of any works in this direction.

2.6.3.7 Algorithm design

Let us first consider an instance of Algorithm 3 by setting the weight $\alpha_t = t$ and let the function $F(\cdot)$ in the definition of the payoff function (2.6) of the game be $F(\cdot) = F_r(\cdot)$ (2.94). Furthermore, let the online learning algorithms OAlg^y and OAlg^x in the game respectively as:

$$
\begin{aligned}
y_t &\leftarrow \operatorname*{argmin}_y \left\{ \alpha_t \ell_{t-1}(y) + \sum_{s=1}^{t-1} \alpha_s \ell_s(y) \right\} &(2.106)\\
&= \operatorname*{argmax}_y \langle \widetilde{x}_t, y \rangle - F_r^*(y) = \nabla F_r(\widetilde{x}_t), &(2.107)
\end{aligned}
$$

where we denote $\widetilde{x}_t := \frac{1}{A_t}(\alpha_t x_{t-1} + \sum_{s=1}^{t-1} \alpha_s x_s)$, and

$$
\hat{x}_t \leftarrow \bar{\Psi}(-\eta \sum_{s=1}^{t} \alpha_s y_s) = \bar{\Psi}(-\eta \sum_{s=1}^{t} \alpha_s \nabla F_r(\widetilde{x}_s)). \tag{2.108}
$$

As we have seen earlier in this chapter, the strategy (2.107) is OPTIMISTICFTL, while the strategy (2.108) can be viewed as a variant of the dual averaging strategy (see e.g. Xiao [280]) but with the learner being prescient, i.e. knows the loss function of the current round before playing an action. By choosing the parameter η and the weighting scheme $\{\alpha_t\}$ appropriately, we can show that the weighted regret is $O(\frac{\hat{L}\log d}{T^2})$. Hence, it will lead to an $O(\frac{\hat{L}\log d}{T^2})$ algorithm for solving the underlying problem $\min_{X \in \Delta_{d_1+d_2}} F_r(X) := f(2rX^{(2)})$.

However, the strategy (2.108) involves computing the expectation, $\bar{\Psi}(\cdot)$, which is hard to achieve in practice. So we propose an unbiased version of it,

$$
x_t \leftarrow \frac{1}{m_t} \sum_{j_t=1}^{m_t} x_{t,j_t} := \frac{1}{m_t} \sum_{j_t=1}^{m_t} \Psi_{u_{j_t}}(-\eta \sum_{s=1}^{t} \alpha_s y_s), \tag{2.109}
$$

where each $u_{j_t} \sim \text{Uni}(\mathbb{S}_d)$.

Theorem 18. *Algorithm 15 is exactly equivalent to Algorithm 3 if $\alpha_t = t$, $F(\cdot) \leftarrow F_r(\cdot)$, and the y-player plays according to (2.107), while the x-player plays according to (2.109). Specifically, there is a following correspondence: $W_t = \bar{x}_t$, $X_t = x_t$, and $Z_t = \tilde{x}_t$, given the same initialization $X_0 = x_0 = \tilde{x}_1$.*

We defer the proof of Theorem 18 to Section 2.7.6. By Theorem 3 and 18, to prove Theorem 17, it suffices to upper-bound the sum of weighted regrets of both players in the game when the x-player plays according to (2.109) and the y-player outputs (2.107). The proof of Theorem 17 is available in Section 2.7.5, which follows the above discussion but has to deal with the case that the x-player plays according to (2.109) instead of the expected one (2.108).

2.6.3.8 Related works

There have been growing research works for projection-free algorithms in recent years (e.g. [161, 114, 253, 58, 20, 114, 289, 82, 264, 50, 203]). When the underlying function is strongly convex or satisfies a notion called quadratic growth in addition to being smooth, there are Frank-Wolfe-like algorithms for the nuclear norm constraint that achieve a better rate than the original Frank-Wolfe method, with a less expensive cost than that of a full-rank SVD (e.g. [8, 97, 73]). However, it is unclear if the algorithms still have the benefit when the function is only smooth but not strongly convex. For the same problem in this work, Garber [98] show that with a warm-start initialization, each iteration of PGD or Accelerated PGD does not need the full-rank SVD computations but a low-rank SVD instead under certain conditions. Garber and Kaplan [102] propose an efficient implementation of Matrix Multiplicative Weight Algorithm [261] that avoids a full-rank eigen-decomposition under certain conditions and enjoys a $O(1/t)$ local convergence rate from a warm-start initialization for the spectrahedron constraint. In this work, we aim at developing an algorithm that avoids SVD computations while achieves a better convergence rate over the nuclear norm ball without the assumption of the strong convexity nor the need of a warm-start initialization.

We notice that in the literature, there are some works suggesting some efforts to *parallelize* the Frank-Wolfe method (e.g. [271, 292, 294, 265]). We want to emphasize that these works are fundamentally different from ours. Wang et al. [271] consider a setting wherein the linear optimization problem of Frank-Wolfe can be decomposed into several smaller ones due to a property called "block-separable" of the variables, which is present in the dual form of structural SVM (Lacoste-Julien et al. [158]), and propose solving them in parallel. The block-separable property does not hold in our problem. Zhang et al., Wan, Tu, and Zhang [292, 265] consider a setting that there is a network of learners and each learner commits an action in each round according to the Online Frank-Wolfe method (Hazan and Kale [128]). The goal is to minimize the sum of regrets of all the learners. So the goal and the notion of parallelization is different from ours. Zheng, Bellet, and Gallinari [294] consider exploiting parallel computing to parallelize the computations of matrix-vector products inside the power iteration, i.e. linear optimization oracle, of Frank-Wolfe for solving problem (2.88), while our work deals with parallelizing the calls to the proposed oracle so that the calls can be made simultaneously. The parallelization is used on a different level. In particular, one can parallelize the internal computations (e.g. matrix-vector products, summations) of computing a single $\Psi_{u.}(\cdot)$ of ours as well. But it is tricky to parallelize the calls to the linear optimization oracle of Frank-Wolfe. The notion of parallelization is different and complementary.

2.7 Detailed proofs

2.7.1 Proof of Theorem 4

Proof. This is a result of the following lemmas.

Definition: *[Definition 12.1 in Rockafellar and Wets [231]] A mapping $T : \mathcal{R}^n \to \mathcal{R}^n$ is called monotone if it has the property that*

$$\langle v_1 - v_0, x_1 - x_0 \rangle \geq 0 \text{ whenever } v_0 \in T(x_0), v_1 \in T(x_1).$$

Moreover, T is maximal monotone if there is no monotone operator that properly contains it.

Lemma 2: *[Theorem 12.17 in Rockafellar and Wets [231]] For a proper, lsc, convex func-*

tion f, ∂f is a maximum monotone operator.

Lemma 3: *[Theorem 12.41 in Rockafellar and Wets [231]] For any maximal monotone mapping T, the set "domain of T" is nearly convex, in the sense that there is a convex set C such that $C \subset$ domain of $T \subset cl(C)$. The same applies to the range of T.*

Therefore, the closure of $\{\partial f(x) | x \in \mathcal{X})\}$ is also convex, because we can define another proper, lsc, convex function $\hat{f}(x)$ such that it is $\hat{f}(x) = f(x)$ if $x \in \mathcal{X}$; otherwise, $\hat{f}(x) = \infty$. Then, the sub-differential of $\hat{f}(x)$ is equal to $\{\partial f(x) | x \in \mathcal{X}\}$. So, we can apply the the lemmas to get the result.

□

2.7.2 Proof of Theorem 7

Note that in the algorithm that we describe below the weights α_t are not predefined but rather depend on the queries of the algorithm. These adaptive weights are explicitly defined in Algorithm 16 which is used by the y-player. Note that Algorithm 16 is equivalent to performing FTL updates over the following loss sequence: $\left\{\tilde{\ell}_t(y) := \alpha_t \ell_t(y)\right\}_{t=1}^{T}$. The x-player plays best response, which only involves the linear optimization oracle.

Algorithm 16 Strongly-Convex Adaptive Follow-the-Leader (SC-AFTL)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Play $y_t \in \mathcal{Y}$
3: Receive a strongly convex loss function $\alpha_t \ell_t(\cdot)$ with $\alpha_t = \frac{1}{\|\nabla \ell_t(y_t)\|^2}$.
4: Update $y_{t+1} = \min_{y \in \mathcal{Y}} \sum_{s=1}^{t} \alpha_y \ell_s(y)$
5: **end for**

Proof. Since the x-player plays best response, $\boldsymbol{\alpha}$-$\textsc{Reg}^x = 0$, we only need to show that the y-player's regret satisfies $\boldsymbol{\alpha}$-$\textsc{Reg}^y \leq O(\exp(-\frac{\lambda B}{L} T))$, which we do next.

We start by defining a function $s(y) := \max_{x \in \mathcal{X}} -x^\top y + f^*(y)$ is a strongly convex function. We are going to show that $s(\cdot)$ is also smooth. We have that

$$\begin{aligned}
&\|\nabla_w s(\cdot) - \nabla_z s(\cdot)\| = \|\arg\max_{x\in\mathcal{X}}(-w^\top w + f^*(w)) - \arg\max_{x\in\mathcal{X}}(-z^\top x + f^*(z))\| \\
&= \|\arg\max_{x\in\mathcal{X}}(-w^\top x) - (\arg\max_{x'\in\mathcal{X}} -z^\top x')\| \leq \frac{2\|w-z\|}{\lambda(\|w\|+\|z\|)} \leq \frac{\|w-z\|}{\lambda B},
\end{aligned} \tag{2.110}$$

where the second to last inequality uses Lemma 16 regarding λ-strongly convex sets, and the last inequality is by assuming the gradient of $\|\nabla f(\cdot)\| \geq B$ and the fact that $w, z \in \mathcal{Y}$ are gradients of f. This shows that $s(\cdot)$ is a smooth function with smoothness constant $L' := \frac{1}{\lambda B}$.

$$\begin{aligned}
T &= \sum_{t=1}^{T} \frac{\|\nabla \ell_t(y_t)^2\|}{\|\nabla \ell_t(y_t)\|^2} \overset{Proposition\ 1}{=} \sum_{t=1}^{T} \frac{\|\nabla s(y_t)\|^2}{\|\nabla \ell_t(y_t)\|^2} \overset{Lemma\ 15}{\leq} \sum_{t=1}^{T} \frac{L'}{\|\nabla \ell_t(y_t)\|^2}(s(y_t) - s(y^*)) \\
&\leq \sum_{t=1}^{T} \frac{L'}{\|\nabla \ell_t(x_t)\|^2}(\ell_t(y_t) - \ell_t(y^*)),
\end{aligned} \tag{2.111}$$

where we denote $y^* := \arg\min_y s(y)$ and the last inequality follows from the fact that $s(y_t) := \ell_t(y_t)$ and $\ell_t(y) = -g(x_t, y) \leq -g(x_y, y) = s(y)$ for any y.

In the following, we will denote c a constant such that $\|\nabla \ell_t(y_t)\| = \|x_t - \nabla f^*(y_t)\| = \|x_t - \bar{x}_{t-1}\| \leq c$. We have

$$\begin{aligned}
T &\leq \sum_{t=1}^{T} \frac{L'}{\|\ell_t(y_t)\|^2}(\ell_t(y_t) - \ell_t(y^*)) \\
&\overset{(a)}{=} \sum_{t=1}^{T} L'(\tilde{\ell}_t(y_t) - \tilde{\ell}_t(y^*)) \\
&\overset{(b)}{\leq} \frac{L \cdot L'}{2} \sum_{t=1}^{T} \frac{\|\nabla \ell_t(y_t)\|^{-2}}{\sum_{s=1}^{t} \|\nabla \ell_s(y_t)\|^{-2}} \\
&\overset{(c)}{\leq} \frac{L \cdot L'}{2} \left(1 + \log(c^2 \sum_{t=1}^{T} \|\nabla \ell_t(y_t)\|^{-2})\right),
\end{aligned}$$

where (a) is by the definition of $\tilde{\ell}_t(\cdot)$, and (b) is shown using Lemma 2 with strong convexity parameter of $\ell_t(\cdot)$ being $\frac{1}{L}$, and (c) is by Lemma 17 so that

$$\sum_{t=1}^{T} \frac{\|\ell_t(y_t)\|^{-2}}{\sum_{s=1}^{t} \|\ell_s(y_s)\|^{-2}} = \sum_{t=1}^{T} \frac{c^2\|\ell_t(y_t)\|^{-2}}{\sum_{s=1}^{t} c^2\|\ell_s(y_s)\|^{-2}} \leq 1 + \log(c^2 \sum_{t=1}^{T} \|\ell_t(y_t)\|^{-2}).$$

Thus, we get

$$c^2 \sum_{t=1}^{T} \|\nabla \ell_t(y_t)\|^{-2} = O(e^{\frac{1}{L \cdot L'} T}) = O(e^{\frac{\lambda B}{L} T}). \tag{2.112}$$

$$\begin{aligned}
\frac{\boldsymbol{\alpha}\text{-}\mathrm{REG}^y}{A_T} &:= \frac{\sum_{t=1}^{T} \alpha_t(\ell_t(y_t) - \ell_t(y^*))}{A_T} \leq \frac{L}{2A_T} \sum_{t=1}^{T} \frac{\|\nabla \ell_t(y_t)\|^{-2}}{\sum_{\tau=1}^{t} \|\nabla \ell_t(y_\tau)\|^{-2}} \\
&\overset{(a)}{\leq} \frac{Lc^2 \left(1 + \log\left(c^2 \sum_{t=1}^{T} \|\nabla \ell_t(y_t)\|^{-2}\right)\right)}{2c^2 \sum_{t=1}^{T} \|\nabla \ell_t(y_t)\|^{-2}} \overset{(b)}{\leq} O(\frac{Lc^2\left(1 + (\frac{\lambda B T}{L})\right)}{e^{\frac{\lambda B}{L} T}}) = O\left(Lc^2 e^{-\frac{\lambda B}{L} T}\right)
\end{aligned} \tag{2.113}$$

where (a) is by Lemma 17, (b) is by (2.112) and the fact that $\frac{1+\log z}{z}$ is monotonically decreasing for $z \geq 1$. This completes the proof.

$\square$

Proposition 1. *For arbitrary y, let $\ell(\cdot) := -g(x_y, \cdot)$. Then $-\nabla_y \ell(\cdot) \in \partial_y s(\cdot)$, where x_y means that the x-player plays x by* BESTRESP$^+$ *after observing the y-player plays y.*

Proof. Consider any point $w \in \mathcal{Y}$,

$$\begin{aligned}
s(w) - s(y) &= g(x_y, y) - g(x_w, w) \\
&= g(x_y, y) - g(x_y, w) + g(x_y, w) - g(x_w, w) \geq g(x_y, y) - g(x_y, w) + 0 \\
&\geq \langle \partial_y g(x_y, y), w - y \rangle = \langle -\nabla_y \ell(y), w - y \rangle
\end{aligned} \tag{2.114}$$

where the first inequality is because that x_w is the best response to w, the second inequality is due to the concavity of $g(x_y, \cdot)$. The overall statement implies that $-\nabla_y \ell(y)$ is a subgradient of s at y. $\square$

Lemma 15. *For any L-smooth convex function $\ell(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$, if $x^* = \operatorname{argmin}_{x\in\mathbb{R}^d} \ell(x)$, then*

$$\|\nabla \ell(x)\|^2 \leq 2L\left(\ell(x) - \ell(x^*)\right), \quad \forall x \in \mathbb{R}^d .$$

Lemma 16. [5] *Denote $x_p = \operatorname{argmax}_{x\in\mathcal{K}}\langle p, x\rangle$ and $x_q = \operatorname{argmax}_{x\in\mathcal{K}}\langle q, x\rangle$, where $p, q \in \mathbb{R}^d$ are any nonzero vectors. If a compact set $\mathcal{K}$ is a λ-strongly convex set, then*

$$\|x_p - x_q\| \leq \frac{2\|p-q\|}{\lambda(\|p\| + \|q\|)}. \tag{2.115}$$

Proof. Polovinkin [222] show that a strongly convex set $\mathcal{K}$ can be written as intersection of some Euclidean balls. Namely,

$$\mathcal{K} = \bigcap_{u:\|u\|_2=1} B_{\frac{1}{\lambda}}\left(x_u - \frac{u}{\lambda}\right),$$

where x_u is defined as $x_u = \operatorname{argmax}_{x\in\mathcal{K}}\langle \frac{u}{\|u\|}, x\rangle$.

Let $x_p = \operatorname{argmax}_{x\in\mathcal{K}}\langle \frac{p}{\|p\|}, x\rangle$ and $x_q = \operatorname{argmax}_{x\in\mathcal{K}}\langle \frac{q}{\|q\|}, x\rangle$. Based on the definition of strongly convex sets, we can see that $x_q \in B_{\frac{1}{\lambda}}(x_p - \frac{p}{\lambda\|p\|})$ and $x_p \in B_{\frac{1}{\lambda}}(x_q - \frac{q}{\lambda\|q\|})$. Therefore,

$$\|x_q - x_p - \frac{p}{\lambda\|p\|}\|^2 \leq \frac{1}{\lambda^2},$$

which leads to

$$\|p\| \cdot \|x_p - x_q\|^2 \leq \frac{2}{\lambda}\langle x_p - x_q, p\rangle. \tag{2.116}$$

Similarly,

$$\|x_p - x_q - \frac{q}{\lambda\|q\|}\|^2 \leq \frac{1}{\lambda^2},$$

which results in

$$\|q\| \cdot \|x_p - x_q\|^2 \leq \frac{2}{\lambda}\langle x_q - x_p, q\rangle. \tag{2.117}$$

[5] Polovinkin [222] discuss the smoothness of the support function on strongly convex sets. Here, we state a more general result.

Summing (2.116) and (2.117), one gets $(\|p\| + \|q\|)\|x_p - x_q\|^2 \leq \frac{2}{\lambda}\langle x_p - x_q, p - q\rangle$. Applying the Cauchy-Schwarz inequality completes the proof. $\square$

Lemma 17. *(Levy [170]) For any non-negative real numbers* $a_1, \ldots, a_n \geq 1$,

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{j=1}^{i} a_j} \leq 1 + \log\left(\sum_{i=1}^{n} a_i\right) .$$

2.7.3 Proof of Theorem 8

Proof. The equivalency of the update follows the proof of Theorem 5. Specifically, we have that the objects on the left in the following equalities correspond to Alg. 3 and those on the right to Alg. 5.

$$x_t = v_t \tag{2.118}$$

$$\bar{x}_t = w_t. \tag{2.119}$$

To analyze the regret of the y-player, we define $\{\hat{y}_t\}$ as the points if the y-player would have played FTL.

$$\begin{aligned} \hat{y}_t &:= \arg\min_y \frac{1}{t-1}\sum_{s=1}^{t-1} \ell_t(y) = \arg\max_y \frac{1}{t-1}\sum_{s=1}^{t-1} \langle x_s, y\rangle - f^*(y) \\ &= \nabla f(\bar{x}_t) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\bar{x}_t) \end{aligned} \tag{2.120}$$

Then, we have that

$$
\begin{aligned}
\overline{\boldsymbol{\alpha}\text{-}\mathrm{REG}}^{y} &= \frac{1}{T}\big(\sum_{t=1}^{T}\ell_t(\hat{y}_t)-\ell_t(y_*)\big)+\frac{1}{T}\big(\sum_{t=1}^{T}\ell_t(y_t)-\ell_t(\hat{y}_t)\big) \\
&\overset{(a)}{\leq} \frac{4LD\log T}{T}+\frac{1}{T}\big(\sum_{t=1}^{T}\ell_t(y_t)-\ell_t(\hat{y}_t)\big) \\
&= \frac{4LD\log T}{T}+\frac{1}{T}\sum_{t=1}^{T}\big(f^*(y_t)-f^*(\hat{y}_t)+\langle x_t,\hat{y}_t-y_t\rangle\big) \\
&\overset{(b)}{\leq} \frac{4LD\log T}{T}+\sum_{t=1}^{T}\frac{1}{T}(L_0+r)\|y_t-\hat{y}_t\| \\
&= \frac{4LD\log T}{T}+\frac{1}{T}(L_0+r)\sum_{t=1}^{T}\|\frac{1}{n}\sum_{i=1}^{n}g_{i,t}-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{x}_t)\| \\
&= \frac{4LD\log T}{T}+\frac{1}{T}(L_0+r)\sum_{t=1}^{T}\|\frac{1}{n}\sum_{i\neq i_t}^{n}\big(g_{i,t}-\nabla f_i(\bar{x}_t)\big)\| \\
&\overset{(c)}{\leq} \frac{4LD\log T}{T}+\frac{L(L_0+r)}{Tn}\sum_{t=1}^{T}\sum_{i\neq i_t}^{n}\|\bar{x}_{\tau_t(i)}-\bar{x}_t\| \\
&\overset{(d)}{\leq} \frac{4LD\log T}{T}+\frac{L(L_0+r)}{Tn}\sum_{t=1}^{T}\sum_{i\neq i_t}^{n}\frac{2nr}{t} \\
&= O(\frac{\max\{LD,L(L_0+r)nr\}\log T}{T}),
\end{aligned}
\tag{2.121}
$$

where (a) is by the regret of FTL (Lemma 2),

$$
\frac{1}{T}\big(\sum_{t=1}^{T}\ell_t(\hat{y}_t)-\ell_t(y_*)\big)\leq\frac{1}{T}\sum_{t=1}^{T}\frac{2\|\nabla\ell_t(\hat{y}_t)\|^2}{\sum_{s=1}^{t}(1/L)}\quad=\frac{4LD\log T}{T},
$$

where we used that $\|\nabla\ell_t(\hat{y}_t)\|^2=\|x_t-\nabla f^*(\hat{y}_t)\|^2=\|x_t-\bar{x}_{t-1}\|^2\leq D$, (b) we assume that the conjugate is L_0-Lipschitz and that $\max_{x\in\mathcal{K}}\|x\|\leq r$, (c) we denote $\tau_t(i)\in[T]$ as the last iteration that i_{th} sample's gradient is computed at t, and (d) is because that

$$\begin{aligned}\|\bar{x}_{\tau_t(i)} - \bar{x}_t\| &= \|\frac{1}{\tau_t(i)}\sum_{s=1}^{\tau_t(i)} x_s - \frac{1}{t}\sum_{s=1}^{t} x_s\| \leq \|\sum_{s=1}^{\tau_t(i)} x_s(\frac{1}{\tau_t(i)} - \frac{1}{t})\| + \|\frac{1}{t}\sum_{s=\tau_t(i)+1}^{t} x_s\| \\ &= \frac{t-\tau_t(i)}{t}\|\bar{x}_{\tau_t(i)}\| + \|\frac{1}{t}\sum_{s=\tau_t(i)+1}^{t} x_s\| \\ &\leq \frac{nr}{t} + \|\frac{1}{t}\sum_{s=\tau_t(i)+1}^{t} x_s\| = \frac{nr}{t} + \frac{t-\tau_t(i)}{t}\|\frac{1}{t-\tau_t(i)}\sum_{s=\tau_t(i)+1}^{t} x_s\| \\ &\leq \frac{2nr}{t}.\end{aligned} \tag{2.122}$$

For the x-player, since it plays BESTRESP$^+$, the regret is non-positive.

Combining the average regrets of both players leads to the result.

□

2.7.4 Proof of Theorem 12

Proof. First, we can bound the norm of the gradient as

$$\|\nabla \ell_t(y_t)\|^2 = \|x_t - \nabla f^*(y_t)\|^2 = \|x_t - \bar{x}_{t-1}\|^2$$

Combining this with Lemma 2 we see that

$$\begin{aligned}\overline{\boldsymbol{\alpha}\text{-REG}}^y[\text{FTL}] &\leq \frac{1}{A_T}\sum_{t=1}^{T}\frac{2\alpha_t^2\|\nabla \ell_t(y_t)\|^2}{\sum_{s=1}^{t}\alpha_s(1/L)} \leq \frac{1}{A_T}\sum_{t=1}^{T}\frac{2\alpha_t^2\|x_t - \bar{x}_{t-1}\|^2}{\sum_{s=1}^{t}\alpha_s(1/L)} \\ &= O(\sum_{\tau=1}^{T}\frac{L\|\bar{x}_{t-1} - x_t\|^2}{A_T}).\end{aligned}$$

On the other hand, the x-player plays OMD$^+$, according to Lemma 11, its regret satisfies

$$\overline{\boldsymbol{\alpha}\text{-REG}}^x \leq \frac{\frac{1}{\gamma}D - \sum_{t=1}^{T}\frac{1}{2\gamma}\|x_{t-1} - x_t\|^2}{A_T} \tag{2.123}$$

Since the distance terms may not cancel out, one can only bound the differences of the distance terms by a constant, which leads to the non-accelerated $O(1/T)$ rate.

$\square$

2.7.5 Proof of Theorem 17

Assume that the spectral norm of the gradient $\|\nabla f(\cdot)\|_2$ over the nuclear-norm ball $\mathcal{NB}_{d_1,d_2}(r)$ satisfies $\|\nabla f(\cdot)\|_2 \leq G$. Then,

$$\begin{aligned} \|\nabla F_r(X)\|_\infty &:= \max\{|\lambda_{\min}(\nabla F_r(X))|, |\lambda_{\max}(\nabla F_r(X))|\} \\ &= \|\nabla f(2rX^{(2)})\|_2 \leq G, \end{aligned} \tag{2.124}$$

where the equality is due to the structure of the gradient matrix (2.96).

Proof. (of Theorem 17) Following the discussion in the main text, Subsection 2.6.3.7, we consider an instance of Algorithm 3 by setting the weight $\alpha_t = t$, the function $F(\cdot)$ in the definition of the payoff function (2.6) of the game as $F(\cdot) \leftarrow F_r(\cdot)$ as defined in (2.94). Furthermore, let OAlg^y and OAlg^x in Algorithm 3 respectively as:

$$y_t \leftarrow \text{argmin}_y \left\{\alpha_t \ell_{t-1}(y) + \textstyle\sum_{s=1}^{t-1} \alpha_s \ell_s(y)\right\} = \nabla F_r(\widetilde{x}_t), \tag{2.125}$$

$$x_t \leftarrow \tfrac{1}{m_t} \textstyle\sum_{j_t=1}^{m_t} \Psi_{u_{j_t}}(-\eta \sum_{s=1}^{t} \alpha_s y_s), \text{ where each } u_{j_t} \sim \text{Uni}(\mathbb{S}_d), \tag{2.126}$$

where $\widetilde{x}_t := \frac{1}{A_t}(\alpha_t x_{t-1} + \sum_{s=1}^{t-1} \alpha_s x_s)$. We also need a *ghost* sequence $\{\hat{x}_s\}_{s=1}^t$ solely used for the analysis,

$$\hat{x}_t \leftarrow \bar{\Psi}(-\eta \sum_{s=1}^{t} \alpha_s y_s) = \bar{\Psi}(-\eta \sum_{s=1}^{t} \alpha_s \nabla F_r(\widetilde{x}_s)) \tag{2.127}$$

and we use (2.107) that $y_s = \nabla F_r(\widetilde{x}_s)$. By Lemma the y-player's regret is

$$\boldsymbol{\alpha}\text{-}\mathrm{REG}^y \leq \hat{L} \textstyle\sum_{t=1}^{T} \frac{\alpha_t^2}{A_t} \|x_{t-1} - x_t\|^2, \tag{2.128}$$

where $\hat{L}$ is the smoothness constant of the underlying function $F_r(\cdot)$.

In the following, we denote the Bregman divergence with the distance generating function $\bar{\psi}(\cdot)$ (defined in the preliminary section):

$$V_C(B) = \bar{\psi}(B) - \bar{\psi}(C) - \langle \bar{\Psi}(C), B - C \rangle, \tag{2.129}$$

for any symmetric matrices $B, C \in \mathcal{S}_d$. Recall that $\bar{\Psi}(\cdot) = \nabla \bar{\psi}(\cdot)$.

Now we are going to analyze the regret of the x-player. But before that, let us analyze the regret if the x-player would have played $\hat{x}_t$. We have

$$\begin{aligned}
\boldsymbol{\alpha}\text{-}\mathrm{REG}^{\hat{x}} &:= \sum_{t=1}^{T} \alpha_t \langle \hat{x}_t - x^*, y_t \rangle \\
&\overset{(a)}{\leq} \sum_{t=1}^{T} \langle \hat{x}_t - x^*, t \nabla F_r(\widetilde{x}_t) \rangle \\
&\overset{(b)}{=} \sum_{t=1}^{T} \frac{1}{\eta} \langle \hat{x}_t - x^*, G_{t-1} - G_t \rangle \\
&\overset{(c)}{\leq} \sum_{t=1}^{T} \frac{1}{\eta} \left(V_{G'}(G_{t-1}) - V_{G'}(G_t) - V_{G_t}(G_{t-1}) \right) \leq \frac{1}{\eta} \left(V_{G'}(G_0) - \sum_{t=1}^{T} V_{G_t}(G_{t-1}) \right) \\
&\overset{(d)}{\leq} \frac{1}{\eta} \left(\log 4d - \sum_{t=1}^{T} V_{G_t}(G_{t-1}) \right) \\
&\overset{(e)}{\leq} \frac{1}{\eta} \left(\log 4d - \sum_{t=1}^{T} \frac{1}{6} \| \hat{x}_t - \hat{x}_{t-1} \|^2 \right),
\end{aligned} \tag{2.130}$$

where (a) is by $\alpha_t = t$ and the y-player strategy (2.125), (b) we define $G_t = G_{t-1} - \eta t \nabla F_r(\widetilde{x}_t)$, (c) we use the well-known three-point inequality:

$$\langle \bar{\Psi}(B_1) - \bar{\Psi}(B_0), B_2 - B_1 \rangle = V_{B_0}(B_2) - V_{B_0}(B_1) - V_{B_1}(B_2), \tag{2.131}$$

and we let $B_1 \leftarrow G_t$, $\hat{x}_t = \bar{\Psi}(G_t)$, $B_2 \leftarrow G_{t-1}$, $x^* = \bar{\Psi}(G')$ and $B_0 \leftarrow G'$ for some symmetric matrix $G' \in \mathcal{S}_d$, (d) we use that $G_0 = 0_d$ and that $V_{G'}(0_d) \leq \log 4d$ for any

$G' \in \mathcal{S}_d$ by Proposition 1 of Carmon et al. [45], (e) we use that $V_{G_t}(G_{t-1}) \geq \frac{1}{6}\|\hat{x}_t - \hat{x}_{t-1}\|^2$ by Proposition 1 of Carmon et al. [45] and Lemma 16 in Krichene [154], as $\hat{x}_t = \bar{\Psi}(G_t)$ and $\hat{x}_{t-1} = \bar{\Psi}(G_{t-1})$.

So the regret of the x-player using strategy (2.126) can be bounded as

$$\begin{aligned} \boldsymbol{\alpha}\text{-}\textsc{Reg}^x := \sum_{t=1}^{T} \alpha_t \langle x_t - x^*, y_t \rangle &= \sum_{t=1}^{T} \alpha_t \langle \hat{x}_t - x^*, y_t \rangle + \alpha_t \langle x_t - \hat{x}_t, y_t \rangle \\ &\overset{(2.130)}{\leq} \tfrac{\log(4d) - \frac{1}{6}\sum_{t=1}^{T} \|\hat{x}_t - \hat{x}_{t-1}\|^2}{\eta} + \textstyle\sum_{t=1}^{T} \alpha_t \langle x_t - \hat{x}_t, y_t \rangle. \end{aligned} \tag{2.132}$$

For the terms $\{\alpha_s \langle x_s - \hat{x}_s, y_s \rangle\}_{s=1}^{t}$, notice that it is a martingale difference sequence. Using the fact that $\alpha_s = s$ and that y_s is a gradient at some point which is bounded, i.e. $\|\nabla F_r(\cdot)\|_\infty \leq G$, see (2.96) and (2.124), we have that $\alpha_s \langle x_{s,j_s}, y_s \rangle \leq \alpha_s \|y_s\|_\infty \leq \alpha_s G$. Hoeffding's lemma implies that $\{\alpha_s \langle x_{s,j_s} - \hat{x}_s, y_s \rangle\}$ is $\alpha_s^2 G^2$-sub-Gaussian, and consequently $\{\alpha_s \langle x_s - \hat{x}_s, y_s \rangle\} = \{\frac{1}{m_s} \sum_{j_s=1}^{m_s} \alpha_s \langle x_{s,j_s} - \hat{x}_s, y_s \rangle\}$ is $\frac{\alpha_s^2 G^2}{m_s}$-sub-Gaussian. The fact that $\xi_s := \alpha_s \langle x_s - \hat{x}_s, y_s \rangle$ is $\frac{\alpha_s^2 G^2}{m_s}$-sub-Gaussian implies that $\max\{Pr(\xi_s \geq \theta), Pr(\xi_s \leq -\theta)\} \leq 2\exp(-\frac{m_s}{2\alpha_s^2 G^2}\theta^2)$. So we can apply a variant of Azuma-Hoeffding inequality (Lemma 18 in Section 2.7.7) to conclude that

$$\sum_{s=1}^{t} \alpha_s \langle x_s - \hat{x}_s, y_s \rangle \leq \sqrt{112 G^2 \log(2/\delta) \sum_{s=1}^{t} \frac{\alpha_s^2}{m_s}}, \tag{2.133}$$

with probability at least $1 - \delta/2$. Therefore, the sum of the weighted average regret of both players is bounded by

$$
\begin{aligned}
&\overline{\boldsymbol{\alpha}\text{-REG}}^{x} + \overline{\boldsymbol{\alpha}\text{-REG}}^{y} \\
&\overset{(2.133),(2.132)}{\leq} \frac{\frac{\log(4d)}{\eta} + \sum_{t=1}^{T}\left(\frac{\alpha_t^2}{A_t}\hat{L}\|x_{t-1}-x_t\|^2 - \frac{1}{6\eta}\|\hat{x}_{t-1}-\hat{x}_t\|^2\right) + \sqrt{112G^2\log(2/\delta)\sum_{t=1}^{T}\frac{\alpha_t^2}{m_t}}}{A_T} \\
&\overset{(a)}{\leq} \frac{\frac{\log(4d)}{\eta} + \sum_{t=1}^{T}\left(\frac{\alpha_t^2}{A_t}3\hat{L}\|\hat{x}_{t-1}-\hat{x}_t\|^2 - \frac{1}{6\eta}\|\hat{x}_{t-1}-\hat{x}_t\|^2\right)}{A_T} \\
&\quad + \frac{\sum_{t=1}^{T}\frac{96\hat{L}\log(4d/\delta)}{m_t} + \sqrt{112G^2\log(2/\delta)\sum_{t=1}^{T}\frac{\alpha_t^2}{m_t}}}{A_T} \\
&\overset{(b)}{\leq} O\left(\frac{\hat{L}\log(d)}{T^2}\right) + \frac{\sum_{t=1}^{T}\frac{96\hat{L}\log(4d/\delta)}{m_t} + \sqrt{112G^2\log(2/\delta)\sum_{t=1}^{T}\frac{t^2}{m_t}}}{T(T+1)/2},
\end{aligned}
\tag{2.134}
$$

where (a) is because $\|x_{t-1}-x_t\|^2 = \|x_{t-1}-\hat{x}_{t-1}+\hat{x}_{t-1}-\hat{x}_t+\hat{x}_t-x_t\|^2 \leq 3(\|x_{t-1}-\hat{x}_{t-1}\|^2 + \|x_t-\hat{x}_t\|^2 + \|\hat{x}_{t-1}-\hat{x}_t\|^2) \overset{(\#)}{\leq} \frac{24\log(4d/\delta)}{m_t} + \frac{24\log(4d/\delta)}{m_{t-1}} + 3\|\hat{x}_{t-1}-\hat{x}_t\|^2$ and the inequality $(\#)$ is due to Theorem 1.6.2 (Matrix Bernstein) of Tropp [258]: $\Pr(\|\frac{1}{m_t}\sum_{j_t=1}^{m_t} x_{t,j_t} - \hat{x}_t\| \geq \theta) \leq 2d\exp(-\frac{m_t\theta^2}{4(1+\frac{\theta}{3})})$, which means that with probability at least $1-\delta/2$, $\|\frac{1}{m_t}\sum_{j_t=1}^{m_t} x_{t,j_t} - \hat{x}_t\| \leq \sqrt{\frac{8\log(4d/\delta)}{m_t}}$ if $\sqrt{\frac{8}{9m_t}\log(4d/\delta)} \leq 1$, and (b) of (2.134) is due to the constraint of $\eta \leq \frac{1}{36\hat{L}}$ so that the distance terms cancel out and that $A_T = \sum_t t$. Thus, by Lemma 14, Theorem 1 we have established the convergence rate.

Since by Theorem 18, Algorithm 15 is exactly equivalent to the instance of Algorithm 3 here, we have completed the proof.

□

2.7.6 Proof of Theorem 18

Proof of Theorem 18. We use proof by induction to show that $W_t = \sum_{s=1}^{t}\frac{\alpha_s}{A_t}x_s = \bar{x}_t$, $X_t = x_t$, and $Z_t = \tilde{x}_t$ for any $t > 0$.

For the base case $t = 1$, we have $W_1 = (1-\beta_1)W_0 + \beta_1 X_1 = X_1 = x_1 = \frac{\alpha_1}{A_1}x_1$, as by the same initialization $Z_1 = X_0 = x_0 = \tilde{x}_1$, one can ensure that $X_1 = x_1$.

Now assume that the one-to-one correspondence holds at $t-1$. We have that $W_t = (1-\beta_t)W_{t-1} + \beta_t x_t = (1-\beta_t)(\sum_{s=1}^{t-1} \frac{\alpha_s}{A_{t-1}} x_s) + \beta_t x_t = (1-\frac{2}{t+1})(\sum_{s=1}^{t-1} \frac{\alpha_s}{\frac{t(t-1)}{2}} x_s) + \beta_t x_t = \sum_{s=1}^{t-1} \frac{\alpha_s}{\frac{t(t+1)}{2}} x_s + \frac{\alpha_t}{A_t} x_t = \sum_{s=1}^{t} \frac{\alpha_s}{A_s} x_s = \bar{x}_t$. On the other hand, we have that $Z_t = (1-\beta_t)W_{t-1} + \beta_t x_{t-1} = (1-\beta_t)(\sum_{s=1}^{t-1} \frac{\alpha_s}{A_{t-1}} x_s) + \beta_t x_{t-1} = (1-\frac{2}{t+1})(\sum_{t=1}^{t-1} \frac{\alpha_t}{\frac{t(t-1)}{2}} x_t) + \beta_t x_{t-1} = \sum_{s=1}^{t-1} \frac{\alpha_s}{\frac{t(t+1)}{2}} x_s + \beta_t x_{t-1} = \sum_{s=1}^{t-1} \frac{\alpha_s}{A_t} x_s + \frac{\alpha_t}{A_t} x_{t-1} = \widetilde{x}_t$. The result implies that $G_t = -\eta \sum_{s=1}^{t} s \nabla F_r(\widetilde{x}_s) = -\eta \sum_{s=1}^{t} \alpha_s y_s$; consequently $X_t = x_t$. We now have completed the proof. □

2.7.7 Some supporting lemmas

Lemma 18. *Let $\xi_1, \xi_2, \ldots, \xi_T$ be a martingale difference sequence with respect to a sequence $\mathcal{F}_1, \ldots, \mathcal{F}_T$, and suppose there are constants $\{b_t\} \geq 1$ and $\{c_t\} > 0$ such that for any $\theta > 0$*

$$\max\{Pr(\xi_t > \theta | \mathcal{F}_1, \ldots, \mathcal{F}_{t-1}), Pr(\xi_t < -\theta | \mathcal{F}_1, \ldots, \mathcal{F}_{t-1})\} \leq b_t \exp(-c_t \theta^2)$$

Then, for any δ, it holds with probability at least $1-\delta$ that

$$\frac{1}{T} \sum_{t=1}^{T} \xi_t \leq \sqrt{\frac{28 \sum_{t=1}^{T} \frac{b_t}{c_t} \log(1/\delta)}{T^2}}.$$

Proof. The lemma's statement is an extension of Theorem 2 in Shamir [237] which considers the case that $b_t = b$ and $c_t = c$ for some numbers $b > 1, c > 0$.

Denote s a positive number.

$$
\begin{aligned}
&Pr(\frac{1}{T}\sum_{t=1}^{T}\xi_t > \epsilon) = Pr\big(\exp(s\sum_{t=1}^{T}\xi_t) > \exp(sT\epsilon)\big) \\
&\overset{(a)}{\leq} \exp(-sT\epsilon)\mathbb{E}[\exp(s\sum_{t=1}^{T}\xi_t)] \leq \exp(-sT\epsilon)\mathbb{E}[\mathbb{E}[\prod_{t=1}^{T}\exp(s\xi_t)|\mathcal{F}_1,\ldots,\mathcal{F}_T]] \\
&\leq \exp(-sT\epsilon)\mathbb{E}\big[\mathbb{E}[\exp(s\xi_T)|\mathcal{F}_1,\ldots,\mathcal{F}_{T-1}]\mathbb{E}[\prod_{t=1}^{T-1}\exp(s\xi_t)|\mathcal{F}_1,\ldots,\mathcal{F}_{T-1}]\big] \\
&\overset{(b)}{\leq} \exp(-sT\epsilon)\exp(\frac{7b_T}{c_T}s^2)\mathbb{E}\big[\mathbb{E}[\prod_{t=1}^{T-1}\exp(s\xi_t)|\mathcal{F}_1,\ldots,\mathcal{F}_{T-1}]\big] \\
&\ldots \\
&\leq \exp(-sT\epsilon + \sum_{t=1}^{T}\frac{7b_t}{c_t}s^2),
\end{aligned}
\tag{2.135}
$$

where (a) is by Markov's inequality and (b) is due to Lemma 19. By setting $s = \frac{T\epsilon}{2\sum_{t=1}^{T}\frac{7b_t}{c_t}}$, we have that $Pr(\frac{1}{T}\sum_{t=1}^{T}\xi_t > \epsilon) \leq \exp(-\frac{T^2\epsilon^2}{28\sum_{t=1}^{T}\frac{b_t}{c_t}})$. Now setting $\delta = \exp(-\frac{T^2\epsilon^2}{28\sum_{t=1}^{T}\frac{b_t}{c_t}})$ and solving ϵ leads to the result. □

Lemma 19. *(Lemma 1 in Shamir [237]) Let ξ be a random variable with $\mathbb{E}[\xi] = 0$, and suppose there exist a constant $b \geq 1$ and a constant $c > 0$ such that for all $\theta > 0$, it holds that*

$$\max\{Pr(\xi \geq \theta), Pr(\xi \leq -\theta)\} \leq b\exp(-c\theta^2).$$

Then for any $s > 0$.

$$\mathbb{E}[\exp(s\xi)] \leq \exp(7bs^2/c).$$

2.8 Conclusion

In this chapter, we present a modular analysis that bridges the online learning/no-regret learning and the classical *offline* convex optimization. The generic scheme also makes designing fast algorithms easier. Simply pitting any two no-regret learning algorithms against each other with an appropriate weighting scheme will lead to an offline convex optimization with a guarantee implied by our meta theorem. We believe our generic acceleration scheme can help to design new algorithms. For example, in online learning there are many adaptive algorithms which enjoy data-dependent regret guarantees and allow a different adaptive learning rate for a different coordinate (e.g. Levy [170] and McMahan [193]). It is interesting to check if our approach of *optimization as iteratively playing a game* can help to design a fast adaptive algorithm for *offline* optimization.

CHAPTER 3
A MODULAR ANALYSIS OF PROVABLE ACCELERATION VIA POLYAK'S MOMENTUM: TRAINING A WIDE RELU NETWORK AND A DEEP LINEAR NETWORK

3.1 Introduction

Momentum methods are very popular for training neural networks in various applications (e.g. [129, 263, 156]). It has been widely observed that the use of momentum helps faster training in deep learning (e.g. [185, 62]). Among all the momentum methods, the most popular one seems to be Polyak's momentum (a.k.a. Heavy Ball momentum) [223], which is the default choice of momentum in PyTorch and Tensorflow. The success of Polyak's momentum in deep learning is widely appreciated and almost all of the recently developed adaptive gradient methods like Adam [151], AMSGrad [229], and AdaBound [190] adopt the use of Polyak's momentum, instead of Nesterov's momentum.

However, despite its popularity, little is known in theory about why Polyak's momentum helps to accelerate training neural networks. Even for convex optimization, problems like strongly convex quadratic problems seem to be one of the few cases that discrete-time Polyak's momentum method provably achieves faster convergence than standard gradient descent (e.g. [168, 116, 108, 115, 183, 184, 40, 234, 88, 275, 90, 70, 241, 132]). On the other hand, the theoretical guarantees of Adam, AMSGrad , or AdaBound are only worse if the momentum parameter β is non-zero and the guarantees deteriorate as the momentum parameter increases, which do not show any advantage of the use of momentum [7]. Moreover, the convergence rates that have been established for Polyak's momentum in several related works [96, 251, 284, 181, 192] do not improve upon those for vanilla gradient descent or vanilla SGD in the worst case. Lessard, Recht, and Packard, Ghadimi, Feyzmahdavian, and Johansson [168, 108] even show negative cases in *convex* optimization that the use of Polyak's momentum results in divergence. Furthermore, Kidambi et al. [150] construct a problem instance for which the momentum method under its optimal tuning is outperformed by other algorithms. A solid understanding of the empirical success of Polyak's momentum in deep learning has eluded researchers for some time.

We begin this chapter by first revisiting the use of Polyak's momentum for the class of strongly convex quadratic problems,

$$\min_{w \in \mathbb{R}^d} \tfrac{1}{2} w^\top \Gamma w + b^\top w, \tag{3.1}$$

where $\Gamma \in \mathbb{R}^{d \times d}$ is a PSD matrix such that $\lambda_{\max}(\Gamma) = \alpha$, $\lambda_{\min}(\Gamma) = \mu > 0$. This is one of the fewknown examples that Polyak's momentum has a provable *globally accelerated* linear rate in the *discrete-time* setting. Yet even for this class of problems existing results only establish an accelerated linear rate in an asymptotic sense and several of them do not have an explicit rate in the non-asymptotic regime (e.g. [223, 168, 196, 227]). Is it possible to prove a non-asymptotic accelerated linear rate in this case? We will return to this question soon.

For general μ-strongly convex, α-smooth, and twice differentiable functions (not necessarily quadratic), denoted as $F^2_{\mu,\alpha}$, Theorem 9 in Polyak [223] shows an asymptotic accelerated linear rate when the iterate is *sufficiently* close to the minimizer so that the landscape can be well approximated by that of a quadratic function. However, the definition of the neighborhood was not very precise in the paper. In this work, we show a locally accelerated linear rate under a quantifiable definition of the neighborhood.

Furthermore, we provably show that Polyak's momentum helps to achieve a faster convergence for training two neural networks, compared to vanilla GD. The first is training a one-layer ReLU network. Over the past few years there have appeared an enormous number of works considering training a one-layer ReLU network, provably showing convergence results for vanilla (stochastic) gradient descent (e.g. [173, 144, 176, 76, 77, 10, 244, 297,

Algorithm 17 Gradient descent with Polyak's momentum [223] (Equivalent Version 1)

Required: Step size parameter η and momentum parameter β.
Init: $w_0 \in \mathbb{R}^d$ and $M_{-1} = 0 \in \mathbb{R}^d$.
for $t = 0$ to T **do**
  Given current iterate w_t, obtain gradient $\nabla \ell(w_t)$.
  Update momentum $M_t = \beta M_{t-1} + \nabla \ell(w_t)$.
  Update iterate $w_{t+1} = w_t - \eta M_t$.
end for

Algorithm 18 Gradient descent with Polyak's momentum [223] (Equivalent Version 2)

Required: step size η and momentum parameter β.
Init: $w_0 = w_{-1} \in \mathbb{R}^d$
for $t = 0$ to T **do**
  Given current iterate w_t, obtain gradient $\nabla \ell(w_t)$.
  Update iterate $w_{t+1} = w_t - \eta \nabla \ell(w_t) + \beta(w_t - w_{t-1})$.
end for

17, 138, 164, 54, 218, 35, 51, 254, 243, 21, 175, 121, 64, 298, 80, 65, 273, 285, 84, 247, 53]), as well as for other algorithms (e.g. [291, 278, 39, 295, 106, 31, 166, 221]). However, we are not aware of any theoretical works that study the momentum method in neural net training except the work Krichene, Caluyay, and Halder [155]. These authors show that SGD with Polyak's momentum (a.k.a. stochastic Heavy Ball) with infinitesimal step size, i.e. $\eta \to 0$, for training a one-hidden-layer network with an infinite number of neurons, i.e. $m \to \infty$, converges to a stationary solution. However, the theoretical result does not show a faster convergence by momentum. In this work we consider the discrete-time setting and nets with finitely many neurons. We provide a non-asymptotic convergence rate of Polyak's momentum, establishing a concrete improvement relative to the best-known rates for vanilla gradient descent.

Our setting of training a ReLU network follows the same framework as previous results, including [76, 17, 244]. Specifically, we study training a one-hidden-layer ReLU neural net of the form,

$$\mathcal{N}_W^{\mathrm{ReLU}}(x) := \tfrac{1}{\sqrt{m}} \textstyle\sum_{r=1}^{m} a_r \sigma(\langle w^{(r)}, x \rangle), \tag{3.2}$$

where $\sigma(z) := z \cdot \mathbb{1}\{z \geq 0\}$ is the ReLU activation, $w^{(1)}, \ldots, w^{(m)} \in \mathbb{R}^d$ are the weights of m neurons on the first layer, $a_1, \ldots, a_m \in \mathbb{R}$ are weights on the second layer, and $\mathcal{N}_W^{\text{ReLU}}(x) \in \mathbb{R}$ is the output predicted on input x. Assume n number of samples $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ is given. Following [76, 17, 244], we define a Gram matrix $H \in \mathbb{R}^{n \times n}$ for the weights W and its expectation $\bar{H} \in \mathbb{R}^{n \times n}$ over the random draws of $w^{(r)} \sim N(0, I_d) \in \mathbb{R}^d$ whose (i, j) entries are defined as follows,

$$\begin{aligned} H(W)_{i,j} &= \sum_{r=1}^{m} \frac{x_i^\top x_j}{m} \mathbb{1}\{\langle w^{(r)}, x_i \rangle \geq 0 \;\&\; \langle w^{(r)}, x_j \rangle \geq 0\} \\ \bar{H}_{i,j} &:= \mathop{\mathbb{E}}_{w^{(r)}} [x_i^\top x_j \mathbb{1}\{\langle w^{(r)}, x_i \rangle \geq 0 \;\&\; \langle w^{(r)}, x_j \rangle \geq 0\}]. \end{aligned} \tag{3.3}$$

The matrix $\bar{H}$ is also called a neural tangent kernel (NTK) matrix in the literature (e.g. [138, 283, 28]). Assume that the smallest eigenvalue $\lambda_{\min}(\bar{H})$ is strictly positive and certain conditions about the step size and the number of neurons are satisfied. Previous works Du et al., Song and Yang [76, 244] show a linear rate of vanilla gradient descent, while we show an accelerated linear rate [1] of gradient descent with Polyak's momentum. As far as we are aware, our result is the first acceleration result of training an over-parametrized ReLU network.

The second result is training a deep linear network. The deep linear network is a canonical model for studying optimization and deep learning, and in particular for understanding gradient descent (e.g. [238, 233, 135]), studying the optimization landscape (e.g. [154, 163]), and establishing the effect of implicit regularization (e.g. [199, 143, 174, 226, 16, 112, 120, 191]). In this work, following Du and Hu [74], Hu, Xiao, and Pennington [135], we study training a L-layer linear network of the form,

$$\mathcal{N}_W^{L\text{-linear}}(x) := \frac{1}{\sqrt{m^{L-1} d_y}} W^{(L)} W^{(L-1)} \cdots W^{(1)} x, \tag{3.4}$$

[1] We borrow the term "accelerated linear rate" from the convex optimization literature [211], because the result here has a resemblance to those results in convex optimization, even though the neural network training is a non-convex problem.

where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ is the weight matrix of the layer $l \in [L]$, and $d_0 = d$, $d_L = d_y$ and $d_l = m$ for $l \neq 1, L$. Therefore, except the first layer $W^{(1)} \in \mathbb{R}^{m \times d}$ and the last layer $W^{(L)} \in \mathbb{R}^{d_y \times m}$, all the intermediate layers are $m \times m$ square matrices. The scaling $\frac{1}{\sqrt{m^{L-1} d_y}}$ is necessary to ensure that the network's output at the initialization $\mathcal{N}^{L\text{-linear}}_{W_0}(x)$ has the same size as that of the input x, in the sense that $\mathbb{E}[\|\mathcal{N}^{L\text{-linear}}_{W_0}(x)\|^2] = \|x\|^2$, where the expectation is taken over some appropriate random initialization of the network (see e.g. [74, 135]). Hu, Xiao, and Pennington [135] show vanilla gradient descent with orthogonal initialization converges linearly and the required width of the network m is independent of the depth L, while we show an accelerated linear rate of Polyak's momentum and the width m is also independent of L. To our knowledge, this is the first acceleration result of training a deep linear network.

A careful reader may be tempted by the following line of reasoning: a deep linear network (without activation) is effectively a simple linear model, and we already know that a linear model with the squared loss gives a quadratic objective for which Polyak's momentum exhibits an accelerated convergence rate. But this intuition, while natural, is not quite right: it is indeed nontrivial even to show that vanilla gradient descent provides a linear rate on deep linear networks [135, 74, 238, 14, 125, 277, 299], as the optimization landscape is non-convex. Existing works show that under certain assumptions, all the local minimum are global [154, 163, 288, 188, 296, 125]. These results are not sufficient to explain the linear convergence of momentum, let alone the acceleration; see Section 3.6 for an empirical result.

Similarly, it is known that under the NTK regime the output of the ReLU network can be approximated by a linear model (e.g. [134]). However, this result alone neither implies a global convergence of any algorithm nor characterizes the optimization landscape. Liu, Zhu, and Belkin [180], Liu, Zhu, and Belkin [179] establish an interesting connection between solving an over-parametrized non-linear system of equations and solving the classical linear system. They show that for smooth and twice differentiable activation, the optimization landscape of an over-parametrized network satisfies a (non-convex) notion called the Polyak-Lokasiewicz (PL) condition [224], i.e. $\frac{1}{2}\|\nabla \ell(w)\|^2 \geq \mu\left(\ell(w) - \ell(w_*)\right)$, where w_* is a global minimizer and $\mu > 0$. It is not clear whether their result can be extended to ReLU activation, however, and the existing result of [66] for the discrete-time Polyak's momentum under the PL condition does not give an accelerated rate nor is it better than that of vanilla GD. Aujol, Dossal, and Rondepierre [19] show a *variant* of Polyak's momentum method having an accelerated rate in a *continuous-time* limit for a problem that satisfies PL and has a unique global minimizer. It is unclear if their result is applicable to our problem. Therefore, showing the advantage of training the ReLU network and the deep linear network by using existing results of Polyak's momentum can be difficult.

To summarize, our contributions in the present work include

- In convex optimization, we show an accelerated linear rate in the non-asymptotic sense for solving the class of the strongly convex quadratic problems via Polyak's momentum (Theorem 25). We also provide an analysis of the accelerated local convergence for the class of functions in $F^2_{\mu,\alpha}$ (Theorem 28 in Section 3.5.9). We establish a technical result (Theorem 23) that helps to obtain these non-asymptotic rates.

- In non-convex optimization, we show accelerated linear rates of the discrete-time Polyak's momentum for training an over-parametrized ReLU network and a deep linear network. (Theorems 26 and 27)

Furthermore, we will develop a modular analysis to show all the results in this work. We identify conditions and propose a meta theorem of acceleration when the momentum method exhibits a certain dynamic, which can be of independent interest. We show that when applying Polyak's momentum for these problems, the induced dynamics exhibit a form where we can directly apply our meta theorem.

3.2 Preliminaries

Throughout this work, $\|\cdot\|_F$ represents the Frobenius norm and $\|\cdot\|_2$ represents the spectral norm of a matrix, while $\|\cdot\|$ represents l_2 norm of a vector. We also denote $\otimes$ the Kronecker product, $\sigma_{\max}(\cdot) = \|\cdot\|_2$ and $\sigma_{\min}(\cdot)$ the largest and the smallest singular value of a matrix respectively.

For the case of training neural networks, we will consider minimizing the squared loss

$$\ell(W) := \tfrac{1}{2} \textstyle\sum_{i=1}^{n} \left(y_i - \mathcal{N}_W(x_i)\right)^2, \tag{3.5}$$

where $x_i \in \mathbb{R}^d$ is the feature vector, $y_i \in \mathbb{R}^{d_y}$ is the label of sample i, and there are n number of samples. For training the ReLU network, we have $\mathcal{N}_W(\cdot) := \mathcal{N}_W^{\text{ReLU}}(\cdot)$, $d_y = 1$, and $W := \{w^{(r)}\}_{r=1}^m$, while for the deep linear network, we have $\mathcal{N}_W(\cdot) := \mathcal{N}_W^{L\text{-linear}}(\cdot)$, and W represents the set of all the weight matrices, i.e. $W := \{W^{(l)}\}_{l=1}^L$. The notation A^k represents the k_{th} matrix power of A.

3.2.1 Prior result of Polyak's momentum

Algorithm 17 and Algorithm 18 show two equivalent presentations of gradient descent with Polyak's momentum. Given the same initialization, one can show that Algorithm 17 and Algorithm 18 generate exactly the same iterates during optimization.

Let us briefly describe a prior acceleration result of Polyak's momentum. The recursive dynamics of Poylak's momentum for solving the strongly convex quadratic problems (3.1) can be written as

$$\begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} = \underbrace{\begin{bmatrix} I_d - \eta\Gamma + \beta I_d & -\beta I_d \\ I_d & 0_d \end{bmatrix}}_{:=A} \cdot \begin{bmatrix} w_{t-1} - w_* \\ w_{t-2} - w_* \end{bmatrix}, \tag{3.6}$$

where w_* is the unique minimizer. By a recursive expansion, one can get

$$\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \| \leq \|A^t\|_2 \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|. \tag{3.7}$$

Hence, it suffices to control the spectral norm of the matrix power $\|A^t\|_2$ for obtaining a convergence rate. In the literature, this is achieved by using Gelfand's formula.

Theorem 19. *(Gelfand [107]; see also Foucart [89]) (Gelfand's formula) Let A be a $d \times d$ matrix. Define the spectral radius $\rho(A) := \max_{i \in [d]} |\lambda_i(A)|$, where $\lambda_i(\cdot)$ is the i_{th} eigenvalue. Then, there exists a non-negative sequence $\{\epsilon_t\}$ such that $\|A^t\|_2 = (\rho(A) + \epsilon_t)^t$ and* $\lim_{t \to \infty} \epsilon_t = 0$.

We remark that there is a lack of the convergence rate of ϵ_t in Gelfand's formula in general.

Denote $\kappa := \alpha/\mu$ the condition number. One can control the spectral radius $\rho(A)$ as $\rho(A) \leq 1 - \frac{2}{\sqrt{\kappa}+1}$ by choosing η and β appropriately, which leads to the following result.

Theorem 20. *(Polyak [223]; see also [168, 227, 196]) Gradient descent with Polyak's momentum with the step size $\eta = \frac{4}{(\sqrt{\mu}+\sqrt{\alpha})^2}$ and the momentum parameter $\beta = \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^2$ has*

$$\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \| \leq (1 - \frac{2}{\sqrt{\kappa}+1} + \epsilon_t)^t \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|,$$

where ϵ_t is a non-negative sequence that goes to zero.

That is, when $t \to \infty$, Polyak's momentum has the $(1 - \frac{2}{\sqrt{\kappa}+1})$ rate, which has a better dependency on the condition number κ than the $1 - \Theta(\frac{1}{\kappa})$ rate of vanilla gradient descent. A concern is that the bound is not quantifiable for a finite t. On the other hand, we are aware of a different analysis that leverages Chebyshev polynomials instead of Gelfand's formula (e.g. [178]), which manages to obtain a $t(1 - \Theta(\frac{1}{\sqrt{\kappa}}))^t$ convergence rate. So the accelerated linear rate is still obtained in an asymptotic sense. Theorem 9 in Can, Gürbüzbalaban, and

Zhu [40] shows a rate $\max\{\bar{C}_1, t\bar{C}_2\}(1-\Theta(\frac{1}{\sqrt{\kappa}})^t)$ for some constants $\bar{C}_1$ and $\bar{C}_2$ under the same choice of the momentum parameter and the step size as Theorem 20. However, for a large t, the dominant term could be $t(1-\Theta(\frac{1}{\sqrt{\kappa}})^t)$. In this work, we aim at obtaining a bound that (I) holds for a wide range of values of the parameters, (II) has a dependency on the squared root of the condition number $\sqrt{\kappa}$, (III) is quantifiable in each iteration and is better than the rate $t(1-\Theta(\frac{1}{\sqrt{\kappa}}))^t$.

Finally, we remark that, to our knowledge, the class of the strongly convex quadratic problems is one of the only known examples that Polyak's momentum has a provable *accelerated linear rate* in terms of the *global convergence* in the *discrete-time* setting. For general smooth, strongly convex, and differentiable functions, a linear rate of the global convergence via discrete-time Polyak's momentum is shown by Ghadimi, Feyzmahdavian, and Johansson [108] and Shi et al. [241]. However, the rate is not an accelerated rate and is not better than that of the vanilla gradient descent.

3.2.2 (One-layer ReLU network) Settings and Assumptions

The ReLU activation is not differentiable at zero. So for solving (3.5), we will replace the notion of gradient in Algorithm 17 and 18 with subgradient $\frac{\partial \ell(W_t)}{\partial w_t^{(r)}} := \frac{1}{\sqrt{m}} \sum_{i=1}^n \big(\mathcal{N}_{W_t}(x_i) - y_i\big) a_r \cdot \mathbb{1}[\langle w_t^{(r)}, x_i \rangle \geq 0] x_i$ and update the neuron r as $w_{t+1}^{(r)} = w_t^{(r)} - \eta \frac{\partial \ell(W_t)}{\partial w_t^{(r)}} + \beta\big(w_t^{(r)} - w_{t-1}^{(r)}\big)$.

As described in the introduction, we assume that the smallest eigenvalue of the Gram matrix $\bar{H} \in \mathbb{R}^{n \times n}$ is strictly positive, i.e. $\lambda_{\min}(\bar{H}) > 0$. We will also denote the largest eigenvalue of the Gram matrix $\bar{H}$ as $\lambda_{\max}(\bar{H})$ and denote the condition number of the Gram matrix as $\kappa := \frac{\lambda_{\max}(\bar{H})}{\lambda_{\min}(\bar{H})}$. Du et al. [76] show that the strict positiveness assumption is indeed mild. Specifically, they show that if no two inputs are parallel, then the least eigenvalue is strictly positive. Panigrahi, Shetty, and Goyal [219] were able to provide a quantitative lower bound under certain conditions. Following the same framework of Du et al. [76], we consider that each weight vector $w^{(r)} \in \mathbb{R}^d$ is initialized according to the normal distribution, i.e. $w^{(r)} \sim N(0, I_d)$, and each $a_r \in R$ is sampled from the Rademacher distribution, i.e. $a_r = 1$ with probability 0.5; and $a_r = -1$ with probability 0.5. We also assume $\|x_i\| \leq 1$ for all samples i. As the previous works (e.g. [173, 144, 76]), we consider only training the first layer $\{w^{(r)}\}$ and the second layer $\{a_r\}$ is fixed throughout the iterations. We will denote $u_t \in \mathbb{R}^n$ whose i_{th} entry is the network's prediction for sample i, i.e. $u_t[i] := \mathcal{N}^{\text{ReLU}}_{W_t}(x_i)$ in iteration t and denote $y \in \mathbb{R}^n$ the vector whose i_{th} element is the label of sample i. The following theorem is a prior result due to Du et al. [76].

Theorem 21. *(Theorem 4.1 in Du et al. [76]) Assume that $\lambda := \lambda_{\min}(\bar{H})/2 > 0$ and that $w_0^{(r)} \sim N(0, I_d)$ and a_r uniformly sampled from $\{-1, 1\}$. Set the number of nodes $m = \Omega(\lambda^{-4} n^6 \delta^{-3})$ and the constant step size $\eta = O(\frac{\lambda}{n^2})$. Then, with probability at least $1 - \delta$ over the random initialization, vanilla gradient descent, i.e. Algorithm 17& 18 with $\beta = 0$, has*

$$\|u_t - y\|^2 \leq (1 - \eta\lambda)^t \cdot \|u_0 - y\|^2.$$

Later Song and Yang [244] improve the network size m to $m = \Omega(\lambda^{-4} n^4 \log^3(n/\delta))$. Wu, Du, and Ward [279] provide an improved analysis over Du et al. [76], which shows that the step size η of vanilla gradient descent can be set as $\eta = \frac{1}{c_1 \lambda_{\max}(\bar{H})}$ for some quantity $c_1 > 0$. The result in turn leads to a convergence rate $(1 - \frac{1}{c_2 \kappa})$ for some quantity $c_2 > 0$. However, the quantities c_1 and c_2 are not universal constants and actually depend on the problem parameters $\lambda_{\min}(\bar{H})$, n, and δ. A question that we will answer in this work is "*Can Polyak's momentum achieve an accelerated linear rate $\left(1 - \Theta(\frac{1}{\sqrt{\kappa}})\right)$, where the factor $\Theta(\frac{1}{\sqrt{\kappa}})$ does not depend on any other problem parameter?*".

3.2.3 (Deep Linear network) Settings and Assumptions

For the case of deep linear networks, we will denote $X := [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$ the data matrix and $Y := [y_1, \ldots, y_n] \in \mathbb{R}^{d_y \times n}$ the corresponding label matrix. We will also denote $\bar{r} := rank(X)$ and the condition number $\kappa := \frac{\lambda_{\max}(X^\top X)}{\lambda_{\bar{r}}(X^\top X)}$. Following Hu, Xiao, and Pennington [135], we will assume that the linear network is initialized by the orthogonal initial-

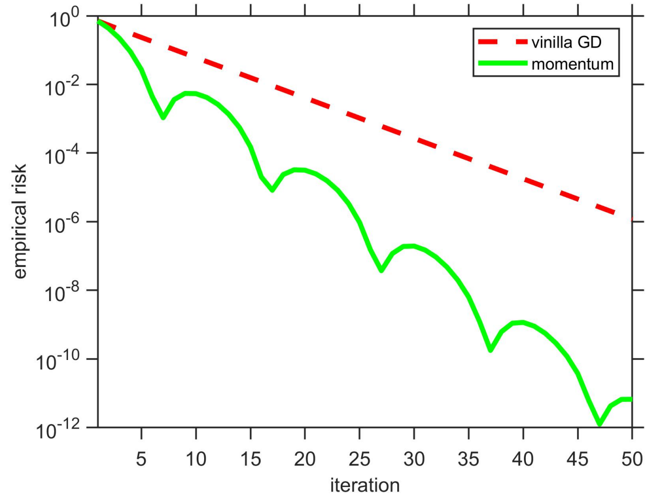


Figure 3.1: Empirical risk $\ell(W_t)$ vs. iteration t. Polyak's momentum accelerates the optimization process of training an over-parametrized one-layer ReLU network. Experimental details are available in Section 3.6.

ization, which is conducted by sampling uniformly from (scaled) orthogonal matrices such that $(W_0^{(1)})^\top W_0^{(1)} = mI_d$, $W_0^{(L)}(W_0^{(L)})^\top = mI_{d_y}$, and $(W_0^{(l)})^\top W_0^{(l)} = W_0^{(l)}(W_0^{(l)})^\top = mI_m$ for layer $2 \leq l \leq L-1$. We will denote $W^{(j:i)} := W_j W_{j-1} \cdots W_i = \Pi_{l=i}^{j} W_l$, where $1 \leq i \leq j \leq L$ and $W^{(i-1:i)} = I$. We also denote the network's output $U := \frac{1}{\sqrt{m^{L-1} d_y}} W^{(L:1)} X \in \mathbb{R}^{d_y \times n}$.

In our analysis, following Du and Hu [74], Hu, Xiao, and Pennington [135], we will further assume that (A1) there exists a W^* such that $Y = W^* X$, $X \in \mathbb{R}^{d \times \bar{r}}$, and $\bar{r} = rank(X)$, which is actually without loss of generality (see e.g. the discussion in Section B of Du and Hu [74]).

Theorem 22. *(Theorem 4.1 in Hu, Xiao, and Pennington [135]) Assume (A1) and the use of the orthogonal initialization. Suppose the width of the deep linear network satisfies* $m \geq C \frac{\|X\|_F^2}{\sigma_{\max}^2(X)} \kappa^2 \big(d_y (1 + \|W_*\|_2^2) + \log(\bar{r}/\delta) \big)$ *and* $m \geq \max\{d_x, d_y\}$ *for some* $\delta \in (0,1)$ *and a sufficiently large constant* $C > 0$*. Set the constant step size* $\eta = \frac{d_y}{2L\sigma_{\max}^2(X)}$*. Then, with probability at least* $1-\delta$ *over the random initialization, vanilla gradient descent, i.e.*

Algorithm 17& 18 with $\beta = 0$, has

$$\|U_t - Y\|_F^2 \leq \left(1 - \Theta(\frac{1}{\kappa})\right)^t \cdot \|U_0 - Y\|_F^2.$$

3.3 Modular Analysis

In this section, we will provide a meta theorem for the following dynamics of the residual vector $\xi_t \in \mathbb{R}^{n_0}$,

$$\begin{bmatrix} \xi_{t+1} \\ \xi_t \end{bmatrix} = \begin{bmatrix} I_{n_0} - \eta H + \beta I_{n_0} & -\beta I_{n_0} \\ I_{n_0} & 0_{n_0} \end{bmatrix} \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} + \begin{bmatrix} \varphi_t \\ 0_{n_0} \end{bmatrix}, \tag{3.8}$$

where η is the step size, β is the momentum parameter, $H \in \mathbb{R}^{n_0 \times n_0}$ is a PSD matrix, $\varphi_t \in \mathbb{R}^{n_0}$ is some vector, and I_{n_0} is the $n_0 \times n_0$-dimensional identity matrix. Note that ξ_t and φ_t depend on the underlying model learned at iteration t, i.e. depend on W_t.

We first show that the residual dynamics of Polyak's momentum for solving all the four problems in this work are in the form of (3.8). The proof of the following lemmas (Lemma 21, 22, and 23) are available in Section 3.5.1.

3.3.1 Realization: Strongly convex quadratic problems

One can easily see that the dynamics of Polyak's momentum (3.6) for solving the strongly convex quadratic problem (3.1) is in the form of (3.8). We thus have the following lemma.

Lemma 20. *Applying Algorithm 17 or Algorithm 18 to solving the class of strongly convex quadratic problems (3.1) induces a residual dynamics in the form of (3.8), where*

$$\xi_t = w_t - w_* \quad \textit{and hence } n_0 = d$$

$$H = \Gamma,$$

$$\varphi_t = 0_d.$$

3.3.2 Realization: Solving $F^2_{\mu,\alpha}$

A similar result holds for optimizing functions in $F^2_{\mu,\alpha}$.

Lemma 21. *Applying Algorithm 17 or Algorithm 18 to minimizing a function $f(w) \in F^2_{\mu,\alpha}$ induces a residual dynamics in the form of (3.8), where*

$$
\begin{aligned}
\xi_t &= w_t - w_* \\
H &= \int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau \\
\varphi_t &= \eta \left(\int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau - \int_0^1 \nabla^2 f\big((1-\tau)w_t + \tau w_*\big) d\tau \right) (w_t - w_*),
\end{aligned}
$$

where $w_* := \arg\min_w f(w)$.

3.3.3 Realization: One-layer ReLU network

More notations: For the analysis, let us define the event $A_{ir} := \{\exists w \in \mathbb{R}^d : \|w - w_0^{(r)}\| \leq R^{\text{ReLU}}, \mathbb{1}\{x_i^\top w_0^{(r)}\} \neq \mathbb{1}\{x_i^\top w \geq 0\}\}$, where $R^{\text{ReLU}} > 0$ is a number to be determined later. The event A_{ir} means that there exists a $w \in \mathbb{R}^d$ which is within the R^{ReLU}-ball centered at the initial point $w_0^{(r)}$ such that its activation pattern of sample i is different from that of $w_0^{(r)}$. We also denote a random set $S_i := \{r \in [m] : \mathbb{1}\{A_{ir}\} = 0\}$ and its complementary set $S_i^\perp := [m] \setminus S_i$.

Lemma 22 below shows that training the ReLU network $\mathcal{N}_W^{\text{-ReLU}}(\cdot)$ via momentum induces the residual dynamics in the form of (3.8).

Lemma 22. *(Residual dynamics of training the ReLU network $\mathcal{N}_W^{ReLU}(\cdot)$) Denote*

$$
(H_t)_{i,j} := H(W_t)_{i,j} = \frac{1}{m} \sum_{r=1}^{m} x_i^\top x_j \times \mathbb{1}\{\langle w_t^{(r)}, x_i \rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j \rangle \geq 0\}.
$$

Applying Algorithm 17 or Algorithm 18 to (3.5) for training the ReLU network $\mathcal{N}_W^{ReLU}(x)$

induces a residual dynamics in the form of (3.8) such that

$$\xi_t[i] = \mathcal{N}^{\text{ReLU}}_{W_t}(x_i) - y_i \quad (\textit{and hence } n_0 = d)$$
$$H = H_0$$
$$\varphi_t = \phi_t + \iota_t,$$

where each element i *of* $\xi_t \in \mathbb{R}^n$ *is the residual error of the sample* i*, and the* i_{th}*-element of* $\phi_t \in \mathbb{R}^n$ *satisfies*

$$|\phi_t[i]| \leq \tfrac{2\eta\sqrt{n}|S_i^{\perp}|}{m}\big(\|u_t - y\| + \beta \textstyle\sum_{s=0}^{t-1} \beta^{t-1-s}\|u_s - y\|\big),$$

and $\iota_t = \eta\,(H_0 - H_t)\,\xi_t \in \mathbb{R}^n$.

3.3.4 Realization: Deep Linear network

Lemma 23 below shows that the residual dynamics due to Polyak's momentum for training the deep linear network is indeed in the form of (3.8). In the lemma, "vec" stands for the vectorization of the underlying matrix in column-first order.

Lemma 23. *(Residual dynamics of training* $\mathcal{N}^{L\text{-linear}}_{W}(\cdot)$*) Denote* $M_{t,l}$ *the momentum term of layer* l *at iteration* t*, which is recursively defined as* $M_{t,l} = \beta M_{t,l-1} + \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}}$*. Denote*

$$H_t := \tfrac{1}{m^{L-1}d_y} \textstyle\sum_{l=1}^{L} [(W_t^{(l-1:1)}X)^\top (W_t^{(l-1:1)}X) \otimes W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top] \in \mathbb{R}^{d_y n \times d_y n}.$$

Applying Algorithm 17 or Algorithm 18 to (3.5) for training the deep linear network $\mathcal{N}^{L\text{-linear}}_{W}(x)$ *induces a residual dynamics in the form of (3.8) such that*

$$\xi_t = \mathit{vec}(U_t - Y) \in \mathbb{R}^{d_y n}, \textit{ and hence } n_0 = d_y n$$
$$H = H_0$$
$$\varphi_t = \phi_t + \psi_t + \iota_t \in \mathbb{R}^{d_y n},$$

where

$$\phi_t = \frac{1}{\sqrt{m^{L-1}d_y}} vec\left(\Phi_t X\right) \text{ with } \Phi_t = \Pi_l(W_t^{(l)} - \eta M_{t,l}) - W_t^{(L:1)} + \eta \sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W_t^{(l-1:1)}$$

$$\psi_t = \frac{1}{\sqrt{m^{L-1}d_y}} vec\left((L-1)\beta W_t^{(L:1)} X + \beta W_{t-1}^{(L:1)} X - \beta \sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)} X\right)$$

$$\iota_t = \eta(H_0 - H_t)\xi_t.$$

3.3.5 A key theorem of bounding a matrix-vector product

Our meta theorem of acceleration will be based on Theorem 23 in the following, which upper-bounds the size of the matrix-vector product of a matrix power A^k and a vector v_0. Compared to Gelfand's formula (Theorem 19), Theorem 23 below provides a better control of the size of the matrix-vector product, since it avoids the dependency on the unknown sequence $\{\epsilon_t\}$. The result can be of independent interest and might be useful for analyzing Polyak's momentum for other problems in future research.

Theorem 23. *Let* $A := \begin{bmatrix} (1+\beta)I_n - \eta H & -\beta I_n \\ I_n & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$. *Suppose that* $H \in \mathbb{R}^{n \times n}$ *is a positive semidefinite matrix. Fix a vector* $v_0 \in \mathbb{R}^n$. *If* β *is chosen to satisfy* $1 \geq \beta > \left(\max\{1 - \sqrt{\eta\lambda_{\min}(H)}, 1 - \sqrt{\eta\lambda_{\max}(H)}\}\right)^2$, *then*

$$\|A^k v_0\| \leq \left(\sqrt{\beta}\right)^k C_0 \|v_0\|, \tag{3.9}$$

where the constant

$$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta, \eta\lambda_{\min}(H)), h(\beta, \eta\lambda_{\max}(H))\}}} \geq 1, \tag{3.10}$$

and the function $h(\beta, z)$ *is defined as* $h(\beta, z) := -\left(\beta - (1-\sqrt{z})^2\right)\left(\beta - (1+\sqrt{z})^2\right)$.

Note that the constant C_0 in Theorem 23 depends on β and ηH. It should be written as $C_0(\beta, \eta H)$ to be precise. However, for the brevity, we will simply denote it as C_0 when the underlying choice of β and ηH is clear from the context. The proof of Theorem 23 is available in Section 3.5.2. Theorem 23 allows us to derive a concrete upper bound of the residual errors in each iteration of momentum, and consequently allows us to show an accelerated linear rate in the non-asymptotic sense. The favorable property of the bound will also help to analyze Polyak's momentum for training the neural networks. As shown later in this chapter, we will need to guarantee the progress of Polyak's momentum in each iteration, which is not possible if we only have a quantifiable bound in the limit. Based on Theorem 23, we have the following corollary. The proof is in Section 3.5.3.

Corollary 3. *Assume that $\lambda_{\min}(H) > 0$. Denote $\kappa := \lambda_{\max}(H)/\lambda_{\min}(H)$. Set $\eta = 1/\lambda_{\max}(H)$ and set $\beta = \left(1 - \frac{1}{2}\sqrt{\eta\lambda_{\min}(H)}\right)^2 = \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^2$. Then, $C_0 \leq 4\sqrt{\kappa}$.*

3.3.6 Meta theorem

Let $\lambda > 0$ be the smallest eigenvalue of the matrix H that appears on the residual dynamics (3.8). Our goal is to show that the residual errors satisfy

$$\left\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \right\| \leq \left(\sqrt{\beta} + \mathbb{1}_{\varphi} C_2\right)^s (C_0 + \mathbb{1}_{\varphi} C_1) \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|, \tag{3.11}$$

where C_0 is the constant defined on (3.10), and $C_1, C_2 \geq 0$ are some constants, $\mathbb{1}_{\varphi}$ is an indicator if any φ_t on the residual dynamics (3.8) is a non-zero vector. For the case of training the neural networks, we have $\mathbb{1}_{\varphi} = 1$.

Theorem 24. *(Meta theorem for the residual dynamics (3.8)) Assume that the step size η and the momentum parameter β satisfying $1 \geq \beta > \left(\max\{1 - \sqrt{\eta\lambda_{\min}(H)}, 1 - \sqrt{\eta\lambda_{\max}(H)}\}\right)^2$ are set appropriately so that (3.11) holds at iteration $s = 0, 1, \ldots, t-1$*

implies that

$$\| \sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix} \| \le \left(\sqrt{\beta} + \mathbb{1}_\varphi C_2\right)^t C_3 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|. \tag{3.12}$$

Then, we have

$$\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \le \left(\sqrt{\beta} + \mathbb{1}_\varphi C_2\right)^t (C_0 + \mathbb{1}_\varphi C_1) \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|, \tag{3.13}$$

holds for all t, where C_0 is defined on (3.10) and $C_1, C_2, C_3 \ge 0$ are some constants satisfying:

$$\begin{aligned} \left(\sqrt{\beta}\right)^t C_0 + \left(\sqrt{\beta} + \mathbb{1}_\varphi C_2\right)^t \mathbb{1}_\varphi C_3 \le& \\ &\left(\sqrt{\beta} + \mathbb{1}_\varphi C_2\right)^t (C_0 + \mathbb{1}_\varphi C_1). \end{aligned} \tag{3.14}$$

Proof. The proof is by induction. At $s = 0$, (3.11) holds since $C_0 \ge 1$ by Theorem 23. Now assume that the inequality holds at $s = 0, 1, \ldots, t-1$. Consider iteration t. Recursively expanding the dynamics (3.8), we have

$$\begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} = A^t \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} + \sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix}. \tag{3.15}$$

By Theorem 23, the first term on the r.h.s. of (3.15) can be bounded by

$$\|A^t \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \le \left(\sqrt{\beta}\right)^t C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \tag{3.16}$$

By assumption, given (3.11) holds at $s = 0, 1, \ldots, t-1$, we have (3.12). Combining (3.12), (3.14), (3.15), and (3.16), we have (3.13) and hence the proof is completed.

□

Remark: As shown in the proof, we need the residual errors be tightly bounded as (3.11) in each iteration. Theorem 23 is critical for establishing the desired result. On the other hand, it would become tricky if instead we use Gelfand's formula or other techniques in the related works that lead to a convergence rate in the form of $O(t\theta^t)$.

3.4 Main results

The important lemmas and theorems in the previous section help to show our main results in the following subsections. The high-level idea to obtain the results is by using the meta theorem (i.e. Theorem 24). Specifically, we will need to show that if the underlying residual dynamics satisfy (3.11) for all the previous iterations, then the terms $\{\varphi_s\}$ in the dynamics satisfy (3.12). This condition trivially holds for the case of the quadratic problems, since there is no such term. On the other hand, for solving the other problems, we need to carefully show that the condition holds. For example, according to Lemma 22, showing acceleration for the ReLU network will require bounding terms like $\|(H_0 - H_s)\xi_s\|$ (and other terms as well), where $H_0 - H_s$ corresponds to the difference of the kernel matrix at two different time steps. By controlling the width of the network, we can guarantee that the change is not too much. A similar result can be obtained for the problem of the deep linear network. The high-level idea is simple but the analysis of the problems of the neural networks can be tedious.

3.4.1 Non-asymptotic accelerated linear rate for solving strongly convex quadratic problems

Theorem 25. *Assume the momentum parameter β satisfies $1 \geq \beta > \left(\max\{1 - \sqrt{\eta\mu}, 1 - \sqrt{\eta\alpha}\}\right)^2$. Gradient descent with Polyak's momentum for solving (3.1) has*

$$\left\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \right\| \leq \left(\sqrt{\beta}\right)^t C_0 \left\| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \right\|, \tag{3.17}$$

where the constant C_0 is defined as

$$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\mu),h(\beta,\eta\alpha)\}}} \geq 1, \tag{3.18}$$

and $h(\beta, z) = -\left(\beta - (1-\sqrt{z})^2\right)\left(\beta - (1+\sqrt{z})^2\right)$. *Consequently, if the step size* $\eta = \frac{1}{\alpha}$ *and the momentum parameter* $\beta = \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^2$, *then it has*

$$\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \| \leq \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^t 4\sqrt{\kappa} \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|. \tag{3.19}$$

Furthermore, if $\eta = \frac{4}{(\sqrt{\mu}+\sqrt{\alpha})^2}$ *and* β *approaches* $\beta \to \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^2$ *from above, then it has a convergence rate approximately* $\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)$ *as* $t \to \infty$.

The convergence rates shown on (3.17) and (3.90) do not depend on the unknown sequence $\{\epsilon_t\}$. Moreover, the rates depend on the squared root of the condition number $\sqrt{\kappa}$. We have hence established a non-asymptotic accelerated linear rate of Polyak's momentum, which helps to show the advantage of Polyak's momentum over vanilla gradient descent in the finite t regime. Our result also recovers the rate $\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)$ asymptotically under the same choices of the parameters as the previous works. The detailed proof can be found in Section 3.5.4, which is actually a trivial application of Lemma 20, Theorem 24, and Corollary 3 with $C_1 = C_2 = C_3 = 0$.

In Section 3.5.9 (Theorem 28), we also provide a local acceleration result for general smooth strongly convex and twice differentiable function $F^2_{\mu,\alpha}$ of the discrete-time Polyak's momentum.

3.4.2 Acceleration for training $\mathcal{N}^{\text{ReLU}}_W(x)$

Before introducing our result, we need the following lemma.

Lemma 24. *[Lemma 3.1 in Du et al. [76] and Song and Yang [244]] Set* $m = \Omega(\lambda^{-2} n^2 \log(n/\delta))$.

Suppose that the neurons $w_0^{(1)}, \ldots, w_0^{(m)}$ are i.i.d. generated by $N(0, I_d)$ initially. Then, with probability at least $1-\delta$, it holds that

$$
\begin{gathered}
\|H_0 - \bar{H}\|_F \leq \frac{\lambda_{\min}(\bar{H})}{4}, \\
\lambda_{\min}\big(H_0\big) \geq \frac{3}{4}\lambda_{\min}(\bar{H}), \qquad \lambda_{\max}\big(H_0\big) \leq \lambda_{\max}(\bar{H}) + \frac{\lambda_{\min}(\bar{H})}{4}.
\end{gathered}
$$

Lemma 24 shows that by the random initialization, with probability $1-\delta$, the least eigenvalue of the Gram matrix $H := H_0$ defined in Lemma 22 is lower-bounded and the largest eigenvalue is also close to $\lambda_{\max}(\bar{H})$. Furthermore, Lemma 24 implies that the condition number of the Gram matrix H_0 at the initialization $\hat{\kappa} := \frac{\lambda_{\max}(H_0)}{\lambda_{\min}(H_0)}$ satisfies

$$
\hat{\kappa} \leq \frac{4}{3}\kappa + \frac{1}{3}, \tag{3.20}
$$

where $\kappa := \frac{\lambda_{\max}(\bar{H})}{\lambda_{\min}(\bar{H})}$.

Theorem 26. *(One-layer ReLU network $\mathcal{N}_W^{ReLU}(x)$) Assume that $\lambda := \frac{3\lambda_{\min}(\bar{H})}{4} > 0$ and that $w_0^{(r)} \sim N(0, I_d)$ and a_r uniformly sampled from $\{-1, 1\}$. Denote $\lambda_{\max} := \lambda_{\max}(\bar{H}) + \frac{\lambda_{\min}(\bar{H})}{4}$ and denote $\hat{\kappa} := \lambda_{max}/\lambda = (4\kappa + 1)/3$. Set a constant step size $\eta = \frac{1}{\lambda_{\max}}$, fix momentum parameter $\beta = \left(1 - \frac{1}{2\hat{\kappa}}\right)^2$, and finally set the number of network nodes $m = \Omega(\lambda^{-4} n^4 \kappa^2 \log^3(n/\delta))$. Then, with probability at least $1-\delta$ over the random initialization, gradient descent with Polyak's momentum satisfies for any t,*

$$
\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \left(1 - \frac{1}{4\sqrt{\hat{\kappa}}}\right)^t \cdot 8\sqrt{\hat{\kappa}} \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|. \tag{3.21}
$$

We remark that $\hat{\kappa}$, which is the condition number of the Gram matrix H_0, is within a constant factor of the condition number of $\bar{H}$ (recall that $\kappa := \frac{\lambda_{\max}(\bar{H})}{\lambda_{\min}(\bar{H})}$). Therefore, Theorem 26 essentially shows an accelerated linear rate $\left(1 - \Theta(\frac{1}{\sqrt{\kappa}})\right)$. The rate has an improved dependency on the condition number, i.e. $\sqrt{\kappa}$ instead of κ, which shows the advantage of Polyak's momentum over vanilla GD when the condition number is large. We believe this is an interesting result, as the acceleration is akin to that in convex optimization, e.g. [211, 241].

Our result also implies that over-parametrization helps acceleration in optimization. To our knowledge, in the literature, there is little theory of understanding why over-parametrization can help training a neural network faster. The only exception that we are aware of is Arora, Cohen, and Hazan [15], which shows that the dynamic of vanilla gradient descent for an over-parametrized objective function exhibits some momentum terms, although their message is very different from ours. The proof of Theorem 26 is in Section 3.5.5.

3.4.3 Acceleration for training $\mathcal{N}_W^{L\text{-linear}}(x)$

Theorem 27. *(Deep linear network $\mathcal{N}_W^{L\text{-linear}}(x)$) Assume (A1) and denote $\lambda := \frac{L\sigma_{\min}^2(X)}{d_y}$. Set a constant step size $\eta = \frac{d_y}{L\sigma_{\max}^2(X)}$, fix momentum parameter $\beta = \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^2$, and finally set a parameter m that controls the width $m \geq C\frac{\kappa^5}{\sigma_{\max}^2(X)}\left(d_y(1 + \|W^*\|_2^2) + \log(\bar{r}/\delta)\right)$ and $m \geq \max\{d_x, d_y\}$ for some constant $C > 0$. Then, with probability at least $1 - \delta$ over the random orthogonal initialization, gradient descent with Polyak's momentum satisfies for any t,*

$$\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \left(1 - \frac{1}{4\sqrt{\kappa}}\right)^t \cdot 8\sqrt{\kappa} \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|. \tag{3.22}$$

Compared with Theorem 22 of Hu, Xiao, and Pennington [135] for vanilla GD, our result clearly shows the acceleration via Polyak's momentum, as it improves the dependency of the condition number to $\sqrt{\kappa}$ (recall that $\kappa := \frac{\sigma_{\max}^2(x)}{\sigma_{\min}^2(x)}$ in this case). Furthermore, the result suggests that the depth does not hurt optimization. Acceleration is achieved for any depth L and the required width m is independent of the depth L as [135, 299] (of vanilla GD). The proof of Theorem 27 is in Section 3.5.7.

3.5 Detailed proofs

3.5.1 Proof of Lemma 21, Lemma 22, and Lemma 23

Lemma 21: *Applying Algorithm 17 or Algorithm 18 to minimizing a function* $f(w) \in F^2_{\mu,\alpha}$ *induces a residual dynamics in the form of (3.8), where*

$$
\begin{aligned}
\xi_t &= w_t - w_* \\
H &= \int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau \\
\varphi_t &= \eta \left(\int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau - \int_0^1 \nabla^2 f\big((1-\tau)w_t + \tau w_*\big) d\tau \right) (w_t - w_*),
\end{aligned}
$$

where $w_* := \arg\min_w f(w)$.

Proof. We have

$$
\begin{aligned}
&\begin{bmatrix} w_{t+1} - w_* \\ w_t - w_* \end{bmatrix} = \begin{bmatrix} I_d + \beta I_d & -\beta I_d \\ I_d & 0_d \end{bmatrix} \cdot \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} + \begin{bmatrix} -\eta \nabla f(w_t) \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} I_d - \eta \int_0^1 \nabla^2 f\big((1-\tau)w_t + \tau w_*\big) d\tau + \beta I_d & -\beta I_d \\ I_d & 0_d \end{bmatrix} \cdot \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \\
&= \begin{bmatrix} I_d - \eta \int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau + \beta I_d & -\beta I_d \\ I_d & 0_d \end{bmatrix} \cdot \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \\
&+ \eta \left(\int_0^1 \nabla^2 f\big((1-\tau)w_0 + \tau w_*\big) d\tau - \int_0^1 \nabla^2 f\big((1-\tau)w_t + \tau w_*\big) d\tau \right) (w_t - w_*),
\end{aligned}
\tag{3.23}
$$

where the second equality is by the fundamental theorem of calculus.

$$
\nabla f(w_t) - \nabla f(w_*) = \left(\int_0^1 \nabla^2 f((1-\tau)w_t + \tau w_*) d\tau \right) (w_t - w_*), \tag{3.24}
$$

and that $\nabla f(w_*) = 0$. $\square$

Lemma 22: *(Residual dynamics of training the ReLU network $\mathcal{N}_W^{ReLU}(\cdot)$) Denote*

$$(H_t)_{i,j} := H(W_t)_{i,j} = \frac{1}{m}\sum_{r=1}^{m} x_i^\top x_j \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j\rangle \geq 0\}.$$

Applying Algorithm 17 or Algorithm 18 to (3.5) for training the ReLU network $\mathcal{N}_W^{ReLU}(x)$ induces a residual dynamics in the form of (3.8) such that

$$\begin{aligned}
\xi_t[i] &= \mathcal{N}_{W_t}^{ReLU}(x_i) - y_i \quad \textit{and hence } n_0 = d \\
H &= H_0 \\
\varphi_t &= \phi_t + \iota_t,
\end{aligned}$$

where each element i of $\xi_t \in \mathbb{R}^n$ is the residual error of the sample i, the i_{th}-element of $\phi_t \in \mathbb{R}^n$ satisfies

$$|\phi_t[i]| \leq \tfrac{2\eta\sqrt{n}|S_i^\perp|}{m}\big(\|u_t - y\| + \beta\textstyle\sum_{s=0}^{t-1}\beta^{t-1-s}\|u_s - y\|\big),$$

and $\iota_t = \eta\,(H_0 - H_t)\,\xi_t \in \mathbb{R}^n$.

Proof. For each sample i, we will divide the contribution to $\mathcal{N}(x_i)$ into two groups.

$$\begin{aligned}
\mathcal{N}(x_i) &= \frac{1}{\sqrt{m}}\sum_{r=1}^{m} a_r\sigma(\langle w^{(r)}, x_i\rangle) \\
&= \frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\sigma(\langle w^{(r)}, x_i\rangle) + \frac{1}{\sqrt{m}}\sum_{r\in S_i^\perp} a_r\sigma(\langle w^{(r)}, x_i\rangle).
\end{aligned} \tag{3.25}$$

To continue, let us recall some notations; the subgradient with respect to $w^{(r)} \in \mathbb{R}^d$ is

$$\frac{\partial L(W)}{\partial w^{(r)}} := \frac{1}{\sqrt{m}}\sum_{i=1}^{n}\big(\mathcal{N}(x_i) - y_i\big)a_r x_i \mathbb{1}\{\langle w^{(r)}, x\rangle \geq 0\}, \tag{3.26}$$

and the Gram matrix H_t whose (i, j) element is

$$H_t[i,j] := \frac{1}{m} x_i^\top x_j \sum_{r=1}^{m} \mathbb{1}\{\langle w_t^{(r)}, x_i \rangle \geq 0 \ \&\ \langle w_t^{(r)}, x_j \rangle \geq 0\}. \tag{3.27}$$

Let us also denote

$$H_t^\perp[i,j] := \frac{1}{m} x_i^\top x_j \sum_{r \in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i \rangle \geq 0 \ \&\ \langle w_t^{(r)}, x_j \rangle \geq 0\}. \tag{3.28}$$

We have that

$$\begin{aligned}
\xi_{t+1}[i] &= \mathcal{N}_{t+1}(x_i) - y_i \\
&\overset{(3.25)}{=} \underbrace{\frac{1}{\sqrt{m}} \sum_{r \in S_i} a_r \sigma(\langle w_{t+1}^{(r)}, x_i \rangle)}_{\text{first term}} + \frac{1}{\sqrt{m}} \sum_{r \in S_i^\perp} a_r \sigma(\langle w_{t+1}^{(r)}, x_i \rangle) - y_i.
\end{aligned} \tag{3.29}$$

For the first term above, we have that

$$\begin{aligned}
&\underbrace{\frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\sigma(\langle w_{t+1}^{(r)}, x_i\rangle)}_{\text{first term}} = \frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\sigma(\langle w_t^{(r)} - \eta\frac{\partial L(W_t)}{\partial w_t^{(r)}} + \beta(w_t^{(r)} - w_{t-1}^{(r)}), x_i\rangle)\\
=&\frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\langle w_t^{(r)} - \eta\frac{\partial L(W_t)}{\partial w_t^{(r)}} + \beta(w_t^{(r)} - w_{t-1}^{(r)}), x_i\rangle \cdot \mathbb{1}\{\langle w_{t+1}^{(r)}, x_i\rangle \geq 0\}\\
\overset{(a)}{=}&\frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\langle w_t^{(r)}, x_i\rangle \cdot \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\} + \frac{\beta}{\sqrt{m}}\sum_{r\in S_i} a_r\langle w_t^{(r)}, x_i\rangle \cdot \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}\\
&-\frac{\beta}{\sqrt{m}}\sum_{r\in S_i} a_r\langle w_{t-1}^{(r)}, x_i\rangle \cdot \mathbb{1}\{\langle w_{t-1}^{(r)}, x_i\rangle \geq 0\} - \eta\frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\langle\frac{\partial L(W_t)}{\partial w_t^{(r)}}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}\\
=&\mathcal{N}_t(x_i) + \beta\big(\mathcal{N}_t(x_i) - \mathcal{N}_{t-1}(x_i)\big) - \frac{1}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_t^{(r)}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}\\
&-\frac{\beta}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_t^{(r)}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\} + \frac{\beta}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_{t-1}^{(r)}, x_i\rangle\mathbb{1}\{\langle w_{t-1}^{(r)}, x_i\rangle \geq 0\}\big)\\
&-\eta\underbrace{\frac{1}{\sqrt{m}}\sum_{r\in S_i} a_r\langle\frac{\partial L(W_t)}{\partial w_t^{(r)}}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}}_{\text{last term}},
\end{aligned} \tag{3.30}$$

where (a) uses that for $r \in S_i$, $\mathbb{1}\{\langle w_{t+1}^{(r)}, x_i\rangle \geq 0\} = \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\} = \mathbb{1}\{\langle w_{t-1}^{(r)}, x_i\rangle \geq 0\}$ as the neurons in S_i do not change their activation patterns. We can further bound (3.30) as

$$\begin{aligned}
\overset{(b)}{=}&\mathcal{N}_t(x_i) + \beta\big(\mathcal{N}_t(x_i) - \mathcal{N}_{t-1}(x_i)\big) - \eta\sum_{j=1}^{n}\big(\mathcal{N}_t(x_j) - y_j\big)H(W_t)_{i,j}\\
&-\frac{\eta}{m}\sum_{j=1}^{n} x_i^{\top}x_j(\mathcal{N}_t(x_j) - y_j)\sum_{r\in S_i^{\perp}}\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j\rangle \geq 0\}\\
&-\frac{1}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_t^{(r)}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\} - \frac{\beta}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_t^{(r)}, x_i\rangle\mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}\\
&+\frac{\beta}{\sqrt{m}}\sum_{r\in S_i^{\perp}} a_r\langle w_{t-1}^{(r)}, x_i\rangle\mathbb{1}\{\langle w_{t-1}^{(r)}, x_i\rangle \geq 0\}\big),
\end{aligned} \tag{3.31}$$

where (b) is due to that

$$
\begin{aligned}
&\underbrace{\frac{1}{\sqrt{m}} \sum_{r\in S_i} a_r \langle \tfrac{\partial L(W_t)}{\partial w_t^{(r)}}, x_i \rangle \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0\}}_{\text{last term}} \\
=&\frac{1}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r\in S_i} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j\rangle \geq 0\} \\
=&\sum_{j=1}^{n} \big(\mathcal{N}_t(x_j) - y_j\big) H(W_t)_{i,j} \\
&\qquad\qquad - \frac{1}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r\in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j\rangle \geq 0\}.
\end{aligned}
\tag{3.32}
$$

Combining (3.29) and (3.31), we have that

$$
\begin{aligned}
\xi_{t+1}[i] = \xi_t[i] &+ \beta\big(\xi_t[i] - \xi_{t-1}[i]\big) - \eta \sum_{j=1}^{n} H_t[i,j]\xi_t[j] \\
&- \frac{\eta}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r\in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_t^{(r)}, x_j\rangle \geq 0\} \\
&+ \frac{1}{\sqrt{m}} \sum_{r\in S_i^\perp} a_r\sigma(\langle w_{t+1}^{(r)}, x_i\rangle) - a_r\sigma(\langle w_t^{(r)}, x_i\rangle) - \beta a_r\sigma(\langle w_t^{(r)}, x_i\rangle) + \beta a_r\sigma(\langle w_{t-1}^{(r)}, x_i\rangle).
\end{aligned}
\tag{3.33}
$$

So we can write the above into a matrix form.

$$
\begin{aligned}
\xi_{t+1} &= (I_n - \eta H_t)\xi_t + \beta(\xi_t - \xi_{t-1}) + \phi_t \\
&= (I_n - \eta H_0)\xi_t + \beta(\xi_t - \xi_{t-1}) + \phi_t + \iota_t,
\end{aligned}
\tag{3.34}
$$

where the i element of $\phi_t \in \mathbb{R}^n$ is defined as

$$\begin{aligned}\phi_t[i] = &-\frac{\eta}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r \in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \ \& \ \langle w_t^{(r)}, x_j\rangle \geq 0\} \\ &+ \frac{1}{\sqrt{m}} \sum_{r \in S_i^\perp} \big\{ a_r \sigma(\langle w_{t+1}^{(r)}, x_i\rangle) - a_r \sigma(\langle w_t^{(r)}, x_i\rangle) - \beta a_r \sigma(\langle w_t^{(r)}, x_i\rangle) + \beta a_r \sigma(\langle w_{t-1}^{(r)}, x_i\rangle) \big\}.\end{aligned} \tag{3.35}$$

Now let us bound $\phi_t[i]$ as follows.

$$\begin{aligned}\phi_t[i] = &-\frac{\eta}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r \in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \ \& \ \langle w_t^{(r)}, x_j\rangle \geq 0\} \\ &+ \frac{1}{\sqrt{m}} \sum_{r \in S_i^\perp} \big\{ a_r \sigma(\langle w_{t+1}^{(r)}, x_i\rangle) - a_r \sigma(\langle w_t^{(r)}, x_i\rangle) - \beta a_r \sigma(\langle w_t^{(r)}, x_i\rangle) + \beta a_r \sigma(\langle w_{t-1}^{(r)}, x_i\rangle) \big\} \\ &\overset{(a)}{\leq} \frac{\eta\sqrt{n}|S_i^\perp|}{m}\|u_t - y\| + \frac{1}{\sqrt{m}} \sum_{r \in S_i^\perp} \big(\|w_{t+1}^{(r)} - w_t^{(r)}\| + \beta \|w_t^{(r)} - w_{t-1}^{(r)}\| \big) \\ &\overset{(b)}{=} \frac{\eta\sqrt{n}|S_i^\perp|}{m}\|u_t - y\| + \frac{\eta}{\sqrt{m}} \sum_{r \in S_i^\perp} \big(\|\sum_{s=0}^{t} \beta^{t-s} \frac{\partial L(W_s)}{\partial w_s^{(r)}}\| + \beta \|\sum_{s=0}^{t-1} \beta^{t-1-s} \frac{\partial L(W_s)}{\partial w_s^{(r)}}\| \big) \\ &\overset{(c)}{\leq} \frac{\eta\sqrt{n}|S_i^\perp|}{m}\|u_t - y\| + \frac{\eta}{\sqrt{m}} \sum_{r \in S_i^\perp} \big(\sum_{s=0}^{t} \beta^{t-s} \|\frac{\partial L(W_s)}{\partial w_s^{(r)}}\| + \beta \sum_{s=0}^{t-1} \beta^{t-1-s} \|\frac{\partial L(W_s)}{\partial w_s^{(r)}}\| \big) \\ &\overset{(d)}{\leq} \frac{\eta\sqrt{n}|S_i^\perp|}{m}\|u_t - y\| + \frac{\eta\sqrt{n}|S_i^\perp|}{m} \big(\sum_{s=0}^{t} \beta^{t-s} \|u_s - y\| + \beta \sum_{s=0}^{t-1} \beta^{t-1-s} \|u_s - y\| \big) \\ &= \frac{2\eta\sqrt{n}|S_i^\perp|}{m} \big(\|u_t - y\| + \beta \sum_{s=0}^{t-1} \beta^{t-1-s} \|u_s - y\| \big),\end{aligned} \tag{3.36}$$

where (a) is because $-\frac{\eta}{m}\sum_{j=1}^{n} x_i^\top x_j (\mathcal{N}_t(x_j) - y_j) \sum_{r \in S_i^\perp} \mathbb{1}\{\langle w_t^{(r)}, x_i\rangle \geq 0 \ \& \ \langle w_t^{(r)}, x_j\rangle \geq$

$0\} \leq \frac{\eta |S_i^{\perp}|}{m} \sum_{j=1}^{n} |\mathcal{N}_t(x_j) - y_j| \leq \frac{\eta \sqrt{n} |S_i^{\perp}|}{m} \|u_t - y\|$, and that $\sigma(\cdot)$ is 1-Lipschitz so that

$$\begin{aligned}
&\frac{1}{\sqrt{m}} \sum_{r \in S_i^{\perp}} \left(a_r \sigma(\langle w_{t+1}^{(r)}, x_i \rangle) - a_r \sigma(\langle w_t^{(r)}, x_i \rangle) \right) \leq \frac{1}{\sqrt{m}} \sum_{r \in S_i^{\perp}} |\langle w_{t+1}^{(r)}, x_i \rangle - \langle w_t^{(r)}, x_i \rangle| \\
&\leq \frac{1}{\sqrt{m}} \sum_{r \in S_i^{\perp}} \|w_{t+1}^{(r)} - w_t^{(r)}\| \|x_i\| \leq \frac{1}{\sqrt{m}} \sum_{r \in S_i^{\perp}} \|w_{t+1}^{(r)} - w_t^{(r)}\|,
\end{aligned}$$

similarly, $\frac{-\beta}{\sqrt{m}} \sum_{r \in S_i^{\perp}} \left(a_r \sigma(\langle w_t^{(r)}, x_i \rangle) - a_r \sigma(\langle w_{t-1}^{(r)}, x_i \rangle) \right) \leq \beta \frac{1}{\sqrt{m}} \sum_{r \in S_i^{\perp}} \|w_t^{(r)} - w_{t-1}^{(r)}\|$, (b) is by the update rule (Algorithm 17), (c) is by Jensen's inequality, (d) is because $|\frac{\partial L(W_s)}{\partial w_s^{(r)}}| = |\frac{1}{\sqrt{m}} \sum_{i=1}^{n} \left(u_s[i] - y_i \right) a_r x_i \mathbb{1}\{x^\top w_t^{(r)} \geq 0\}| \leq \frac{\sqrt{n}}{m} \|u_s - y\|$.

□

Lemma: 23 *(Residual dynamics of training $\mathcal{N}_W^{\text{L-linear}}(\cdot)$) Denote $M_{t,l}$ the momentum term of layer l at iteration t, which is recursively defined as $M_{t,l} = \beta M_{t,l-1} + \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}}$. Denote*

$$H_t := \tfrac{1}{m^{L-1} d_y} \textstyle\sum_{l=1}^{L} [(W_t^{(l-1:1)} X)^\top (W_t^{(l-1:1)} X) \otimes W_t^{(L:l+1)} (W_t^{(L:l+1)})^\top] \in \mathbb{R}^{d_y n \times d_y n}.$$

Applying Algorithm 17 or Algorithm 18 to (3.5) for training the deep linear network $\mathcal{N}_W^{\text{L-linear}}(x)$ induces a residual dynamics in the form of (3.8) such that

$$\begin{aligned}
&\xi_t = vec(U_t - Y) \in \mathbb{R}^{d_y n}, \text{ and hence } n_0 = d_y n \\
&H = H_0 \\
&\varphi_t = \phi_t + \psi_t + \iota_t \in \mathbb{R}^{d_y n},
\end{aligned}$$

where

$$\begin{aligned}
\phi_t &= \frac{1}{\sqrt{m^{L-1} d_y}} vec\left(\Phi_t X\right) \text{ with } \Phi_t = \Pi_l (W_t^{(l)} - \eta M_{t,l}) - W_t^{(L:1)} + \eta \sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W_t^{(l-1:1)} \\
\psi_t &= \frac{1}{\sqrt{m^{L-1} d_y}} vec\left((L-1)\beta W_t^{(L:1)} X + \beta W_{t-1}^{(L:1)} X - \beta \sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)} X \right) \\
\iota_t &= \eta (H_0 - H_t) \xi_t.
\end{aligned}$$

Proof. According to the update rule of gradient descent with Polyak's momentum, we have

$$W_{t+1}^{(L:1)} = \Pi_l \left(W_t^{(l)} - \eta M_{t,l} \right) = W_t^{(L:1)} - \eta \sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W^{(l-1:1)} + \Phi_t, \tag{3.37}$$

where $M_{t,l}$ stands for the momentum term of layer l, which is $M_{t,l} = \beta M_{t,l-1} + \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}} = \sum_{s=0}^{t} \beta^{t-s} \frac{\partial \ell(W_s^{(L:1)})}{\partial W_s^{(l)}}$, and Φ_t contains all the high-order terms (in terms of η), e.g. those with $\eta M_{t,i}$ and $\eta M_{t,j}$, $i \neq j \in [L]$, or higher. Based on the equivalent update expression of gradient descent with Polyak's momentum $-\eta M_{t,l} = -\eta \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}} + \beta(W_t^{(l)} - W_{t-1}^{(l)})$, we can rewrite (3.37) as

$$\begin{aligned}
&W_{t+1}^{(L:1)} \\
&= W_t^{(L:1)} - \eta \sum_{l=1}^{L} W_t^{(L:l+1)} \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}} W_t^{(l-1:1)} + \sum_{l=1}^{L} W_t^{(L:l+1)} \beta (W_t^{(l)} - W_{t-1}^{(l)}) W_t^{(l-1:1)} + \Phi_t \\
&= W_t^{(L:1)} - \eta \sum_{l=1}^{L} W_t^{(L:l+1)} \frac{\partial \ell(W_t^{(L:1)})}{\partial W_t^{(l)}} W_t^{(l-1:1)} + \beta (W_t^{(L:1)} - W_{t-1}^{(L:1)}) + \phi_t \\
&\quad + (L-1)\beta W_t^{(L:1)} + \beta W_{t-1}^{(L:1)} - \beta \sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)}.
\end{aligned} \tag{3.38}$$

Multiplying the above equality with $\frac{1}{\sqrt{m^{L-1} d_y}} X$, we get

$$\begin{aligned}
U_{t+1} = U_t &- \eta \frac{1}{m^{L-1} d_y} \sum_{l=1}^{L} W_t^{(L:l+1)} (W_t^{(L:l+1)})^\top (U_t - Y) (W_t^{(l-1:1)} X)^\top W_t^{(l-1:1)} X \\
&+ \frac{1}{\sqrt{m^{L-1} d_y}} \left((L-1)\beta W_t^{(L:1)} + \beta W_{t-1}^{(L:1)} - \beta \sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)} \right) X \\
&+ \frac{1}{\sqrt{m^{L-1} d_y}} \Phi_t X + \beta (U_t - U_{t-1})
\end{aligned} \tag{3.39}$$

Using $\text{vec}(ACB) = (B^\top \otimes A)\text{vec}(C)$, where $\otimes$ stands for the Kronecker product, we can apply a vectorization of the above equation and obtain

$$\begin{aligned}
&\text{vec}(U_{t+1}) - \text{vec}(U_t) = -\eta H_t \text{vec}(U_t - Y) + \beta\left(\text{vec}(U_t) - \text{vec}(U_{t-1})\right) \\
&\quad + \text{vec}(\frac{1}{\sqrt{m^{L-1}d_y}}\left((L-1)\beta W_t^{(L:1)} + \beta W_{t-1}^{(L:1)} - \beta\sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)}\right) X) \\
&\quad + \frac{1}{\sqrt{m^{L-1}d_y}}\text{vec}(\Phi_t X),
\end{aligned} \tag{3.40}$$

where

$$H_t = \frac{1}{m^{L-1}d_y}\sum_{l=1}^{L}\left[\left((W_t^{(l-1:1)}X)^\top (W_t^{(l-1:1)}X)\right) \otimes W_t^{(L:l+1)} (W_t^{(L:l+1)})^\top\right], \tag{3.41}$$

which is a positive semi-definite matrix.

In the following, we will denote $\xi_t := \text{vec}(U_t - Y)$ as the vector of the residual errors. Also, we denote $\phi_t := \frac{1}{\sqrt{m^{L-1}d_y}}\text{vec}(\Phi_t X)$ with $\Phi_t = \Pi_l(W_t^{(l)} - \eta M_{t,l}) - W_t^{(L:1)} + \eta\sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W_t^{(l-1:1)}$, and
$\psi_t := \text{vec}(\frac{1}{\sqrt{m^{L-1}d_y}}\left((L-1)\beta W_t^{(L:1)} + \beta W_{t-1}^{(L:1)} - \beta\sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)}\right) X)$. Using the notations, we can rewrite (3.40) as

$$\begin{aligned}
\begin{bmatrix} \xi_{t+1} \\ \xi_t \end{bmatrix} &= \begin{bmatrix} I_{d_y n} - \eta H_t + \beta I_{d_y n} & -\beta I_{d_y n} \\ I_{d_y n} & 0_{d_y n} \end{bmatrix} \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} + \begin{bmatrix} \phi_t + \psi_t \\ 0_{d_y n} \end{bmatrix} \\
&= \begin{bmatrix} I_{d_y n} - \eta H_0 + \beta I_{d_y n} & -\beta I_{d_y n} \\ I_{d_y n} & 0_{d_y n} \end{bmatrix} \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} + \begin{bmatrix} \varphi_t \\ 0_{d_y n} \end{bmatrix},
\end{aligned} \tag{3.42}$$

where $\varphi_t = \phi_t + \psi_t + \iota_t \in \mathbb{R}^{d_y n}$ and $I_{d_y n}$ is the $d_y n \times d_y n$-dimensional identity matrix.

$\square$

3.5.2 Proof of Theorem 23

Theorem 23 *Let* $A := \begin{bmatrix} (1+\beta)I_n - \eta H & -\beta I_n \\ I_n & 0 \end{bmatrix} \in \mathbb{R}^{2n\times 2n}$. *Suppose that* $H \in \mathbb{R}^{n\times n}$ *is a positive semidefinite matrix. Fix a vector* $v_0 \in \mathbb{R}^n$. *If* β *is chosen to satisfy* $1 \geq \beta > \big(\max\{1-\sqrt{\eta\lambda_{\min}(H)}, 1-\sqrt{\eta\lambda_{\max}(H)}\}\big)^2$, *then*

$$\|A^k v_0\| \leq \big(\sqrt{\beta}\big)^k C_0 \|v_0\|, \tag{3.43}$$

where the constant

$$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H)), h(\beta,\eta\lambda_{\max}(H))\}}} \geq 1, \tag{3.44}$$

and the function $h(\beta, z)$ *is defined as*

$$h(\beta, z) := -\left(\beta - \left(1-\sqrt{z}\right)^2\right)\left(\beta - \left(1+\sqrt{z}\right)^2\right). \tag{3.45}$$

We would first prove some lemmas for the analysis.

Lemma 25. *Under the assumption of Theorem 23,* A *is diagonalizable with respect to complex field* $\mathbb{C}$ *in* $\mathbb{C}^n$, *i.e.,* $\exists P$ *such that* $A = PDP^{-1}$ *for some diagonal matrix* D. *Furthermore, the diagonal elements of* D *all have magnitudes bounded by* $\sqrt{\beta}$.

Proof. In the following, we will use the notation/operation $\mathrm{Diag}(\cdots)$ to represents a block-diagonal matrix that has the arguments on its main diagonal. Let $U\mathrm{Diag}([\lambda_1,\ldots,\lambda_n])U^*$ be the singular-value-decomposition of H, then

$$A = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} (1+\beta)I_n - \eta\mathrm{Diag}([\lambda_1,\ldots,\lambda_n]) & -\beta I_n \\ I_n & 0 \end{bmatrix} \begin{bmatrix} U^* & 0 \\ 0 & U^* \end{bmatrix}. \tag{3.46}$$

Let $\tilde{U} = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix}$. Then, after applying some permutation matrix $\tilde{P}$, A can be further simplified into

$$A = \tilde{U}\tilde{P}\Sigma\tilde{P}^T\tilde{U}^*, \tag{3.47}$$

where Σ is a block diagonal matrix consisting of n 2-by-2 matrices $\tilde{\Sigma}_i := \begin{bmatrix} 1+\beta-\eta\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$. The characteristic polynomial of $\tilde{\Sigma}_i$ is $x^2 - (1+\beta-\lambda_i)x + \beta$. Hence it can be shown that when $\beta > (1-\sqrt{\eta\lambda_i})^2$ then the roots of polynomial are conjugate and have magnitude $\sqrt{\beta}$. These roots are exactly the eigenvalues of $\tilde{\Sigma}_i \in \mathbb{R}^{2\times 2}$. On the other hand, the corresponding eigenvectors $q_i, \bar{q}_i$ are also conjugate to each other as $\tilde{\Sigma}_i \in \mathbb{R}^{2\times 2}$ is a real matrix. As a result, $\Sigma \in \mathbb{R}^{2n\times 2n}$ admits a block eigen-decomposition as follows,

$$\begin{aligned}\Sigma =& \mathrm{Diag}(\tilde{\Sigma}_i, \ldots, \tilde{\Sigma}_n) \\ =& \mathrm{Diag}(Q_1, \ldots, Q_n)\mathrm{Diag}\left(\begin{bmatrix} z_1 & 0 \\ 0 & \bar{z}_1 \end{bmatrix}, \ldots, \begin{bmatrix} z_n & 0 \\ 0 & \bar{z}_n \end{bmatrix}\right)\mathrm{Diag}(Q_1^{-1}, \ldots, Q_n^{-1}),\end{aligned} \tag{3.48}$$

where $Q_i = [q_i, \bar{q}_i]$ and $z_i, \bar{z}_i$ are eigenvalues of $\tilde{\Sigma}_i$ (they are conjugate by the condition on β). Denote $Q := \mathrm{Diag}(Q_1, \ldots, Q_n)$ and

$$D := \mathrm{Diag}\left(\begin{bmatrix} z_1 & 0 \\ 0 & \bar{z}_1 \end{bmatrix}, \ldots, \begin{bmatrix} z_n & 0 \\ 0 & \bar{z}_n \end{bmatrix}\right). \tag{3.49}$$

By combining (3.47) and (3.48), we have

$$A = P\mathrm{Diag}\left(\begin{bmatrix} z_1 & 0 \\ 0 & \bar{z}_1 \end{bmatrix}, \ldots, \begin{bmatrix} z_n & 0 \\ 0 & \bar{z}_n \end{bmatrix}\right)P^{-1} = PDP^{-1}, \tag{3.50}$$

where

$$P = \tilde{U}\tilde{P}Q, \tag{3.51}$$

by the fact that $\tilde{P}^{-1} = \tilde{P}^T$ and $\tilde{U}^{-1} = \tilde{U}^*$. $\square$

Proof. (of Theorem 23) Now we proceed the proof of Theorem 23. In the following, we denote $v_k := A^k v_0$ (so $v_k = Av_{k-1}$). Let P be the matrix in Lemma 25, and $u_k := P^{-1}v_k$, the dynamic can be rewritten as $u_k = P^{-1}Av_{k-1} = P^{-1}APu_{k-1} = Du_{k-1}$. As D is diagonal, we immediately have

$$\begin{aligned}
& \|u_k\| \le \max_{i\in[n]} |D_{ii}|^k \|u_0\| \\
\Rightarrow\ & \|P^{-1}v_k\| \le \max_{i\in[n]} |D_{ii}|^k \|P^{-1}v_0\| \\
\Rightarrow\ & \sigma_{\min}(P^{-1})\|v_k\| \le \sqrt{\beta}^k \sigma_{\max}(P^{-1})\|v_0\| \quad \text{(Lemma 25.)} \\
\Rightarrow\ & \sigma_{\max}^{-1}(P)\|v_k\| \le \sqrt{\beta}^k \sigma_{\min}^{-1}(P)\|v_0\| \\
\Rightarrow\ & \|v_k\| \le \sqrt{\beta}^k \frac{\sigma_{\max}(P)}{\sigma_{\min}(P)}\|v_0\| \\
\Rightarrow\ & \|v_k\| \le \sqrt{\beta}^k \sqrt{\frac{\lambda_{\max}(PP^*)}{\lambda_{\min}(PP^*)}}\|v_0\|.
\end{aligned} \tag{3.52}$$

Hence, now it suffices to prove upper bound and lower bound of $\lambda_{\max}$ and $\lambda_{\min}$, respectively. By using Lemma 26 in the following, we obtain the inequality of (3.43). We remark that as C_0 is an upper-bound of the squared root of the condition number $\sqrt{\frac{\lambda_{\max}(PP^*)}{\lambda_{\min}(PP^*)}}$, it is lower bounded by 1. $\square$

Lemma 26. *Let P be the matrix in Lemma 25, then we have $\lambda_{\max}(PP^*) \le 2(\beta+1)$ and $\lambda_{\min}(PP^*) \ge \min\{h(\beta, \eta\lambda_{\min}(H)), h(\beta, \eta\lambda_{\max}(H))\}/(1+\beta)$, where*

$$h(\beta, z) = -\left(\beta - \left(1-\sqrt{z}\right)^2\right)\left(\beta - \left(1+\sqrt{z}\right)^2\right). \tag{3.53}$$

Proof. As (3.51) in the proof of Lemma 2, $P = \tilde{U}\tilde{P}\mathrm{Diag}(Q_1, \ldots, Q_n)$. Since $\tilde{U}\tilde{P}$ is unitary, it does not affect the spectrum of P, therefore, it suffices to analyze the eigenvalues of QQ^*, where $Q = \mathrm{Diag}(Q_1, \ldots, Q_n)$. Observe that QQ^* is a block diagonal matrix with blocks $Q_iQ_i^*$, the eigenvalues of it are exactly that of $Q_iQ_i^*$, i.e., $\lambda_{\max}(QQ^*) = \max_{i\in[n]} \lambda_{\max}(Q_iQ_i^*)$ and likewise for the minimum. Recall $Q_i = [q_i, \bar{q}_i]$ consisting of eigenvectors of $\tilde{\Sigma}_i := \begin{bmatrix} 1+\beta-\eta\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$ with corresponding eigenvalues $z_i, \bar{z}_i$. The eigenvalues satisfy

$$z_i + \bar{z}_i = 2\Re z_i = 1 + \beta - \eta\lambda_i, \tag{3.54}$$

$$z_i\bar{z}_i = |z_i|^2 = \beta. \tag{3.55}$$

On the other hand, the eigenvalue equation $\tilde{\Sigma}_i q_i = z_i q_i$ together with (3.54) implies $q_i = [z_i, 1]^T$. Furthermore, $Q_iQ_i^* = q_iq_i^* + \bar{q}_i\bar{q}_i^* = 2\Re q_iq_i^* = 2\Re q_i \Re q_i{}^T + 2\Im q_i \Im q_i{}^T$. Thus,

$$\begin{aligned} Q_iQ_i^* &= 2\Re q_i \Re q_i{}^T + 2\Im q_i \Im q_i{}^T \\ &= 2\left(\begin{bmatrix} \Re z_i \\ 1 \end{bmatrix} \begin{bmatrix} \Re z_i & 1 \end{bmatrix} + \begin{bmatrix} \Im z_i \\ 0 \end{bmatrix} \begin{bmatrix} \Im z_i & 0 \end{bmatrix}\right) \\ &= 2\begin{bmatrix} |z_i|^2 & \Re z_i \\ \Re z_i & 1 \end{bmatrix}. \end{aligned} \tag{3.56}$$

Let the eigenvalues of $Q_iQ_i^*$ be θ_1, θ_2, then by (3.54)-(3.56) we must have

$$\theta_1 + \theta_2 = 2(\beta + 1), \tag{3.57}$$

$$\begin{aligned} \theta_1\theta_2 &= 4\left(\beta - (\frac{1+\beta-\eta\lambda_i}{2})^2\right) \\ &= -\left(\beta - \left(1 - \sqrt{\eta\lambda_i}\right)^2\right)\left(\beta - \left(1 + \sqrt{\eta\lambda_i}\right)^2\right) \geq 0. \end{aligned} \tag{3.58}$$

From (3.57), as both eigenvalues are nonnegative, we deduce that

$$2(1+\beta) \geq \max\{\theta_1, \theta_2\} \geq \beta + 1. \tag{3.59}$$

On the other hand, from (3.57) we also have

$$\begin{aligned}
\min\{\theta_1, \theta_2\} =& \theta_1\theta_2 / \max\{\theta_1, \theta_2\} \\
\geq& -\left(\beta - \left(1 - \sqrt{\eta\lambda_i}\right)^2\right)\left(\beta - \left(1 + \sqrt{\eta\lambda_i}\right)^2\right)/(1+\beta) \\
:=& h(\beta, \eta\lambda_i)/(1+\beta).
\end{aligned} \tag{3.60}$$

Finally, as the eigenvalues of QQ^* are composed of exactly that of $Q_iQ_i^*$, applying the bound of (3.60) to each i we have

$$\begin{aligned}
\lambda_{\min}(PP^*) \geq& \min_{i\in[n]} h(\beta, \eta\lambda_i)/(1+\beta) \\
\geq& \min\{h(\beta, \eta\lambda_{\min}(H)), h(\beta, \eta\lambda_{\max}(H))\}/(1+\beta),
\end{aligned} \tag{3.61}$$

where the last inequality follows from the facts that $\lambda_{\min}(H) \leq \lambda_i \leq \lambda_{\max}(H)$ and h is concave quadratic function of of λ in which the minimum must occur at the boundary. $\square$

3.5.3 Proof of Corollary 3

Corollary 3 *Assume that* $\lambda_{\min}(H) > 0$. *Denote* $\kappa := \lambda_{\max}(H)/\lambda_{\min}(H)$. *Set* $\eta = 1/\lambda_{\max}(H)$ *and set* $\beta = \left(1 - \frac{1}{2}\sqrt{\eta\lambda_{\min}(H)}\right)^2 = \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^2$. *Then,* $C_0 \leq \max\{4, 2\sqrt{\kappa}\} \leq 4\sqrt{\kappa}$.

Proof. For notation brevity, in the following, we let $\mu := \lambda_{\min}(H)$ and $\alpha := \lambda_{\max}(H)$.

Recall that $h(\beta, z) = -\left(\beta - (1-\sqrt{z})^2\right)\left(\beta - (1+\sqrt{z})^2\right)$. We have

$$\begin{aligned} h(\beta, \eta\mu) &= -\left((1-\frac{1}{2}\sqrt{\eta\mu})^2 - (1-\sqrt{\eta\mu})^2\right)\left((1-\frac{1}{2}\sqrt{\eta\mu})^2 - (1+\sqrt{\eta\mu})^2\right) \\ &= 3\left(\sqrt{\eta\mu} - \frac{3}{4}\eta\mu\right)\left(\sqrt{\eta\mu} + \frac{1}{4}\eta\mu\right) = 3\left(\frac{1}{\sqrt{\kappa}} - \frac{3}{4\kappa}\right)\left(\frac{1}{\sqrt{\kappa}} + \frac{1}{4\kappa}\right) \end{aligned} \tag{3.62}$$

and

$$\begin{aligned} h(\beta, \eta\alpha) &= -\left((1-\frac{1}{2}\sqrt{\eta\mu})^2 - (1-\sqrt{\eta\alpha})^2\right)\left((1-\frac{1}{2}\sqrt{\eta\mu})^2 - (1+\sqrt{\eta\alpha})^2\right) \\ &= \left(2\sqrt{\eta\alpha} - \sqrt{\eta\mu} - \eta\alpha + \frac{1}{4}\eta\mu\right)\left(\sqrt{\eta\mu} + 2\sqrt{\eta\alpha} + \eta\alpha - \frac{1}{4}\eta\mu\right) \\ &= \left(1 - \frac{1}{\sqrt{\kappa}} + \frac{1}{4\kappa}\right)\left(3 + \frac{1}{\sqrt{\kappa}} - \frac{1}{4\kappa}\right). \end{aligned} \tag{3.63}$$

We can simplify it to get that $h(\beta, \eta\alpha) = 3 - \frac{2}{\sqrt{\kappa}} - \frac{1}{2\kappa} + \frac{1}{2\kappa^{3/2}} - \frac{1}{16\kappa^2} \geq 0.5$.

Therefore, we have

$$\begin{aligned} &\frac{\sqrt{2}(\beta+1)}{\sqrt{h(\beta,\eta\mu)}} = \frac{\sqrt{2}(\beta+1)}{\sqrt{3\eta\mu(1-\frac{1}{2}\sqrt{\eta\mu} - \frac{3}{16}\eta\mu)}} = \frac{\sqrt{2}(\beta+1)}{\sqrt{3(1-\frac{1}{2}\sqrt{\eta\mu} - \frac{3}{16}\eta\mu)}}\sqrt{\kappa} \\ &\leq \frac{1}{\sqrt{(1-\frac{1}{2}-\frac{3}{16})}}\sqrt{\kappa} \leq 2\sqrt{\kappa}, \end{aligned} \tag{3.64}$$

where we use $\eta\mu = \frac{1}{\kappa}$. On the other hand, $\frac{\sqrt{2}(\beta+1)}{\sqrt{h(\beta,\eta\alpha)}} \leq 4$. We conclude that

$$C_0 = \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\mu), h(\beta,\eta\alpha)\}}} \leq \max\{4, 2\sqrt{\kappa}\} \leq 4\sqrt{\kappa}. \tag{3.65}$$

□

3.5.4 Proof of Theorem 25

Theorem 25 *Assume the momentum parameter β satisfies $1 \geq \beta > \left(\max\{1-\sqrt{\eta\mu}, 1-\sqrt{\eta\alpha}\}\right)^2$. Gradient descent with Polyak's momentum has*

$$\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \| \leq \left(\sqrt{\beta}\right)^t C_0 \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|, \tag{3.66}$$

where the constant

$$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\mu), h(\beta,\eta\alpha)\}}}, \tag{3.67}$$

and $h(\beta, z) = -\left(\beta - (1-\sqrt{z})^2\right)\left(\beta - (1+\sqrt{z})^2\right)$. Consequently, if the step size $\eta = \frac{1}{\alpha}$ and the momentum parameter $\beta = \left(1-\sqrt{\eta\mu}\right)^2$, then it has

$$\| \begin{bmatrix} w_t - w_* \\ w_{t-1} - w_* \end{bmatrix} \| \leq \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^t 4\sqrt{\kappa} \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|. \tag{3.68}$$

Furthermore, if $\eta = \frac{4}{(\sqrt{\mu}+\sqrt{\alpha})^2}$ and β approaches $\beta \to \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^2$ from above, then it has a convergence rate approximately $\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)$ as $t \to \infty$.

Proof. The result (3.66) and (3.68) is due to a trivial combination of Lemma 20, Theorem 24, and Corollary 3.

On the other hand, set $\eta = \frac{4}{(\sqrt{\mu}+\sqrt{\alpha})^2}$, the lower bound on β becomes $\left(\max\{1-\sqrt{\eta\mu}, 1-\sqrt{\eta\alpha}\}\right)^2 = \left(1-\frac{2}{\sqrt{\kappa}+1}\right)^2$. Since the rate is $r = \lim_{t\to\infty} \frac{1}{t}\log(\sqrt{\beta}^{t+1} C_0) = \sqrt{\beta}$, setting $\beta \downarrow \left(1-\frac{2}{\sqrt{\kappa}+1}\right)^2$ from above leads to the rate of $\left(1-\frac{2}{\sqrt{\kappa}+1}\right)$. Formally, it is straightforward to show that $C_0 = \Theta\left(1/\sqrt{\beta - (1-\frac{2}{1+\sqrt{\kappa}})^2}\right)$, hence, for any β converges to $(1-\frac{2}{\sqrt{\kappa}+1})^2$ slower than inverse exponential of κ, i.e., $\beta = (1-\frac{2}{\sqrt{\kappa}+1})^2 + (\frac{1}{\kappa})^{o(t)}$, we have $r = 1 - \frac{2}{\sqrt{\kappa}+1}$.

□

3.5.5 Proof of Theorem 26

We will need some supporting lemmas in the following for the proof. In the following analysis, we denote $C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H)),h(\beta,\eta\lambda_{\max}(H))\}}}$, where $h(\beta,\cdot)$ is defined in Theorem 23 and $H = H_0$ whose (i,j) entry is $(H_0)_{i,j} := H(W_0)_{i,j} = \frac{1}{m}\sum_{r=1}^m x_i^\top x_j \mathbb{1}\{\langle w_0^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_0^{(r)}, x_j\rangle \geq 0\}$, as defined in Lemma 22. In the following, we also denote $\beta = (1-\frac{1}{2}\sqrt{\eta\lambda})^2 := \beta_*^2$. We summarize the notations in Table 3.1.

Notation	definition (or value)	meaning
$\mathcal{N}_W^{\mathrm{ReLU}}(x)$	$\mathcal{N}_W^{\mathrm{ReLU}}(x) := \frac{1}{\sqrt{m}}\sum_{r=1}^m a_r\sigma(\langle w^{(r)}, x\rangle)$	the ReLU network's output given x
$\bar{H}$	$\bar{H}_{i,j} := \mathbb{E}_{w^{(r)}}[x_i^\top x_j \mathbb{1}\{\langle w^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w^{(r)}, x_j\rangle \geq 0\}]$.	the expectation of the Gram matrix
H_0	$:= H(W_0)_{i,j}$ $= \frac{1}{m}\sum_{r=1}^m x_i^\top x_j \mathbb{1}\{\langle w_0^{(r)}, x_i\rangle \geq 0 \;\&\; \langle w_0^{(r)}, x_j\rangle \geq 0\}$	the Gram matrix at the initialization
$\lambda_{\min}(\bar{H})$	$\lambda_{\min}(\bar{H}) > 0$ (by assumption)	the least eigenvalue of $\bar{H}$.
$\lambda_{\max}(\bar{H})$		the largest eigenvalue of $\bar{H}$
κ	$\kappa := \lambda_{\max}(\bar{H})/\lambda_{\min}(\bar{H})$	the condition number of $\bar{H}$
λ	$\lambda := \frac{3}{4}\lambda_{\min}(\bar{H})$	(a lower bound of) the least eigenvalue of H_0.
$\lambda_{\max}$	$\lambda_{\max} := \lambda_{\max}(\bar{H}) + \frac{\lambda_{\min}(\bar{H})}{4}$	(an upper bound of) the largest eigenvalue of H_0.
$\hat{\kappa}$	$\hat{\kappa} := \frac{\lambda_{\max}}{\lambda} = \frac{4}{3}\kappa + \frac{1}{3}$	the condition number of H_0.
η	$\eta = 1/\lambda_{\max}$	step size
β	$\beta = (1-\frac{1}{2}\sqrt{\eta\lambda})^2 = (1-\frac{1}{2\sqrt{\hat{\kappa}}})^2 := \beta_*^2$	momentum parameter
β_*	$\beta_* = \sqrt{\beta} = 1-\frac{1}{2}\sqrt{\eta\lambda}$	squared root of β
θ	$\theta = \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1-\frac{1}{4}\sqrt{\eta\lambda} = 1-\frac{1}{4\sqrt{\hat{\kappa}}}$	the convergence rate
C_0	$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H_0)),h(\beta,\eta\lambda_{\max}(H_0))\}}}$	the constant used in Theorem 23

Table 3.1: Summary of the notations for proving Theorem 26.

Lemma 27. *Suppose that the neurons $w_0^{(1)},\ldots,w_0^{(m)}$ are i.i.d. generated by $N(0,I_d)$ initially. Then, for any set of weight vectors $W_t := \{w_t^{(1)},\ldots,w_t^{(m)}\}$ that satisfy for any $r\in[m]$, $\|w_t^{(r)} - w_0^{(r)}\| \leq R^{ReLU} := \frac{\lambda}{1024nC_0}$, it holds that*

$$\|H_t - H_0\|_F \leq 2nR^{ReLU} = \frac{\lambda}{512C_0},$$

with probability at least $1-n^2\cdot\exp(-mR^{ReLU}/10)$.

Proof. This is an application of Lemma 3.2 in Song and Yang [244]. □

Lemma 27 shows that if the distance between the current iterate W_t and its initialization W_0 is small, then the distance between the Gram matrix $H(W_t)$ and $H(W_0)$ should also be small. Lemma 27 allows us to obtain the following lemma, which bounds the size of φ_t (defined in Lemma 22) in the residual dynamics.

Lemma 28. *Following the setting as Theorem 26, denote* $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. *Suppose that* $\forall i \in [n], |S_i^{\perp}| \leq 4mR^{\mathit{ReLU}}$ *for some constant* $R^{\mathit{ReLU}} := \frac{\lambda}{1024nC_0} > 0$. *If we have (I) for any* $s \leq t$, *the residual dynamics satisfies* $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \leq \theta^s \cdot \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$, *for some constant* $\nu > 0$, *and (II) for any* $r \in [m]$ *and any* $s \leq t$, $\|w_s^{(r)} - w_0^{(r)}\| \leq R^{\mathit{ReLU}}$, *then* ϕ_t *and* ι_t *in Lemma 22 satisfies*

$$\|\phi_t\| \leq \frac{\sqrt{\eta\lambda}}{16}\theta^t\nu\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|, \textit{ and } \|\iota_t\| \leq \frac{\eta\lambda}{512}\theta^t\nu\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|.$$

Consequently, φ_t *in Lemma 22 satisfies*

$$\|\varphi_t\| \leq \left(\frac{\sqrt{\eta\lambda}}{16} + \frac{\eta\lambda}{512}\right)\theta^t\nu\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|.$$

Proof. Denote $\beta_* := 1 - \frac{1}{2}\sqrt{\eta\lambda}$ and $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. We have by Lemma 22

$$
\begin{aligned}
\|\phi_t\| = \sqrt{\sum_{i=1}^{n} \phi_t[i]^2} &\leq \sqrt{\sum_{i=1}^{n} \big(\frac{2\eta\sqrt{n}|S_i^{\perp}|}{m}\big(\|\xi_t\| + \beta \sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\|\xi_\tau\|\big)\big)^2} \\
&\overset{(a)}{\leq} 8\eta n R^{\text{ReLU}} \big(\|\xi_t\| + \beta \sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\|\xi_\tau\|\big) \\
&\overset{(b)}{\leq} 8\eta n R^{\text{ReLU}} \left(\theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| + \beta \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} \theta^\tau \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \right) \\
&\overset{(c)}{=} 8\eta n R^{\text{ReLU}} \left(\theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| + \beta_*^2 \nu C_0 \sum_{\tau=0}^{t-1} \beta_*^{2(t-1-\tau)} \theta^\tau \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \right) \\
&\overset{(d)}{\leq} 8\eta n R^{\text{ReLU}} \left(\theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| + \beta_*^2 \nu C_0 \theta^{t-1} \sum_{\tau=0}^{t-1} \theta^{t-1-\tau} \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \right) \\
&\leq 8\eta n R^{\text{ReLU}} \theta^t (1 + \beta_* \sum_{\tau=0}^{t-1} \theta^\tau) \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\leq 8\eta n R^{\text{ReLU}} \theta^t (1 + \frac{\beta_*}{1-\theta}) \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(e)}{\leq} \frac{\sqrt{\eta\lambda}}{16} \theta^t \nu \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|,
\end{aligned}
\tag{3.69}
$$

where (a) is by $|S_i^{\perp}| \leq 4mR^{\text{ReLU}}$, (b) is by induction that $\|\xi_t\| \leq \theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$ as $u_0 = u_{-1}$, (c) uses that $\beta = \beta_*^2$, (d) uses $\beta_* \leq \theta$, (e) uses $1 + \frac{\beta_*}{1-\theta} \leq \frac{2}{1-\theta} \leq \frac{8}{\sqrt{\eta\lambda}}$ and $R^{\text{ReLU}} := \frac{\lambda}{1024 n C_0}$.

Now let us switch to bound $\|\iota_t\|$.

$$
\|\iota_t\| \leq \eta \|H_0 - H_t\|_2 \|\xi_t\| \leq \frac{\eta\lambda}{512 C_0} \theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|, \tag{3.70}
$$

where we uses Lemma 27 that $\|H_0 - H_t\|_2 \le \frac{\lambda}{512 C_0}$ and the induction that $\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \| \le \theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$.

□

The assumption of Lemma 28, $\forall i \in [n], |S_i^{\perp}| \le 4mR^{\text{ReLU}}$ only depends on the initialization. Lemma 30 shows that it holds with probability at least $1 - n \cdot \exp(-mR^{\text{ReLU}})$.

Lemma 29. *Following the setting as Theorem 26, denote $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. Suppose that the initial error satisfies $\|\xi_0\|^2 = O(n \log(m/\delta) \log^2(n/\delta))$. If for any $s < t$, the residual dynamics satisfies $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \le \theta^s \cdot \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$, for some constant $\nu > 0$, then*

$$\|w_t^{(r)} - w_0^{(r)}\| \le R^{\textit{ReLU}} := \frac{\lambda}{1024 n C_0}.$$

Proof. We have

$$\begin{aligned}
\|w_{t+1}^{(r)} - w_0^{(r)}\| &\overset{(a)}{\le} \eta \sum_{s=0}^{t} \|M_s^{(r)}\| \overset{(b)}{=} \eta \sum_{s=0}^{t} \| \sum_{\tau=0}^{s} \beta^{s-\tau} \frac{\partial L(W_\tau)}{\partial w_\tau^{(r)}} \| \le \eta \sum_{s=0}^{t} \sum_{\tau=0}^{s} \beta^{s-\tau} \| \frac{\partial L(W_\tau)}{\partial w_\tau^{(r)}} \| \\
&\overset{(c)}{\le} \eta \sum_{s=0}^{t} \sum_{\tau=0}^{s} \beta^{s-\tau} \frac{\sqrt{n}}{\sqrt{m}} \|y - u_\tau\| \\
&\overset{(d)}{\le} \eta \sum_{s=0}^{t} \sum_{\tau=0}^{s} \beta^{s-\tau} \frac{\sqrt{2n}}{\sqrt{m}} \theta^\tau \nu C_0 \|y - u_0\| \\
&\overset{(e)}{\le} \frac{\eta\sqrt{2n}}{\sqrt{m}} \sum_{s=0}^{t} \frac{\theta^s}{1-\theta} \nu C_0 \|y - u_0\| \le \frac{\eta\sqrt{2n}}{\sqrt{m}} \left(\frac{\nu C_0}{(1-\theta)^2} \right) \|y - u_0\| \\
&\overset{(f)}{=} \frac{\eta\sqrt{2n}}{\sqrt{m}} \left(\frac{16 \nu C_0}{\eta\lambda} \right) \|y - u_0\| \\
&\overset{(g)}{=} \frac{\eta\sqrt{2n}}{\sqrt{m}} \left(\frac{16 \nu C_0}{\eta\lambda} \right) O(\sqrt{n \log(m/\delta) \log^2(n/\delta)}) \\
&\overset{(h)}{\le} \frac{\lambda}{1024 n C_0},
\end{aligned} \tag{3.71}$$

where (a), (b) is by the update rule of momentum, which is $w_{t+1}^{(r)} - w_t^{(r)} = -\eta M_t^{(r)}$, where $M_t^{(r)} := \sum_{s=0}^{t} \beta^{t-s} \frac{\partial L(W_s)}{\partial w_s^{(r)}}$, (c) is because $\|\frac{\partial L(W_s)}{\partial w_s^{(r)}}\| = \| \sum_{i=1}^{n} (y_i - u_s[i]) \frac{1}{\sqrt{m}} a_r x_i \cdot \mathbb{1}\{\langle w_s^{(r)}, x\rangle \geq 0\}\| \leq \frac{1}{\sqrt{m}} \sum_{i=1}^{n} |y_i - u_s[i]| \leq \frac{\sqrt{n}}{\sqrt{m}} \|y - u_s\|$, (d) is by $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \leq \theta^s \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$ (e) is because that $\beta = \beta_*^2 \leq \theta^2$, (f) we use $\theta := (1 - \frac{1}{4}\sqrt{\eta\lambda})$, so that $\frac{1}{(1-\theta)^2} = \frac{16}{\eta\lambda}$, (g) is by that the initial error satisfies $\|y - u_0\|^2 = O(n \log(m/\delta) \log^2(n/\delta))$, and (h) is by the choice of the number of neurons $m = \Omega(\lambda^{-4} n^4 C_0^4 \log^3(n/\delta)) = \Omega(\lambda^{-4} n^4 \kappa^2 \log^3(n/\delta))$, as $C_0 = \Theta(\sqrt{\kappa})$ by Corollary 3.

The proof is completed.

□

Lemma 29 basically says that if the size of the residual errors is bounded and decays over iterations, then the distance between the current iterate W_t and its initialization W_0 is well-controlled. The lemma will allows us to invoke Lemma 27 and Lemma 28 when proving Theorem 26. The proof of Lemma 29 is in Section 3.5.5. The assumption of Lemma 29, $\|\xi_0\|^2 = O(n \log(m/\delta) \log^2(n/\delta))$, is satisfied by the random initialization with probability at least $1 - \delta/3$ according to Lemma 31 .

Lemma 30. *(Claim 3.12 of Song and Yang [244]) Fix a number* $R_1 \in (0,1)$. *Recall that* $S_i^{\perp}$ *is a random set defined in Subsection 3.3.3. With probability at least* $1 - n \cdot \exp(-mR_1)$, *we have that for all* $i \in [n]$,

$$|S_i^{\perp}| \leq 4mR_1.$$

A similar lemma also appears in Du et al. [76]. Lemma 30 says that the number of neurons whose activation patterns for a sample i could change during the execution is only a small faction of m if R_1 is a small number, i.e. $|S_i^{\perp}| \leq 4mR_1 \ll m$.

Lemma 31. *(Claim 3.10 in Song and Yang [244]) Assume that* $w_0^{(r)} \sim N(0, I_d)$ *and* a_r

uniformly sampled from $\{-1, 1\}$*. For* $0 < \delta < 1$*, we have that*

$$\|y - u_0\|^2 = O(n \log(m/\delta) \log^2(n/\delta)),$$

with probability at least $1 - \delta$*.*

3.5.6 Proof of Theorem 26

Proof. (of Theorem 26) Denote $\lambda := \frac{3}{4}\lambda_{\min}(\bar{H}) > 0$. Lemma 24 shows that λ is a lower bound of $\lambda_{\min}(H)$ of the matrix H defined in Lemma 22. Also, denote $\beta_* := 1 - \frac{1}{2}\sqrt{\eta\lambda}$ (note that $\beta = \beta_*^2$) and $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. In the following, we let $\nu = 2$ in Lemma 28, 29, and let $C_1 = C_3 = C_0$ and $C_2 = \frac{1}{4}\sqrt{\eta\lambda}$ in Theorem 24. The goal is to show that $\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \theta^t 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$ for all t by induction. To achieve this, we will also use induction to show that for all iterations s,

$$\forall r \in [m], \|w_s^{(r)} - w_0^{(r)}\| \leq R^{\text{ReLU}} := \tfrac{\lambda}{1024nC_0}, \tag{3.72}$$

which is clear true in the base case $s = 0$.

By Lemma 22, 24, 27, 28, Theorem 24, and Corollary 3, it suffices to show that given $\left\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \right\| \leq \theta^s 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$ and (3.72) hold at $s = 0, 1, \ldots, t-1$, one has

$$\begin{aligned} \|\textstyle\sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix}\| \quad &\leq \quad \theta^t C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|, & (3.73) \\ \forall r \in [m], \|w_t^{(r)} - w_0^{(r)}\| \quad &\leq \quad R^{\text{ReLU}} := \tfrac{\lambda}{1024nC_0}, & (3.74) \end{aligned}$$

where the matrix A and the vector φ_t are defined in Lemma 22. The inequality (3.73) is the required condition for using the result of Theorem 24, while the inequality (3.74) helps us to show (3.73) through invoking Lemma 28 to bound the terms $\{\varphi_s\}$ as shown in the following.

We have

$$\begin{aligned}
&\|\sum_{s=0}^{t-1} A^{t-s-1}\begin{bmatrix}\varphi_s\\ 0\end{bmatrix}\| \overset{(a)}{\leq} \sum_{s=0}^{t-1}\beta_*^{t-s-1}C_0\|\varphi_s\| \\
&\overset{(b)}{\leq}\left(\frac{\sqrt{\eta\lambda}}{16}+\frac{\eta\lambda}{512}\right)2C_0\|\begin{bmatrix}\xi_0\\ \xi_{-1}\end{bmatrix}\|\left(\sum_{s=0}^{t-1}\beta_*^{t-1-s}\theta^s\right) \\
&\overset{(c)}{\leq}\left(\frac{1}{2}+\frac{1}{64}\sqrt{\eta\lambda}\right)\theta^{t-1}C_0\|\begin{bmatrix}\xi_0\\ \xi_{-1}\end{bmatrix}\| \overset{(d)}{\leq}\theta^t C_0\|\begin{bmatrix}\xi_0\\ \xi_{-1}\end{bmatrix}\|,
\end{aligned} \tag{3.75}$$

where (a) uses Theorem 23, (b) is due to Lemma 28, Lemma 30,(c) is because $\sum_{s=0}^{t-1}\beta_*^{t-1-s}\theta^s = \theta^{t-1}\sum_{s=0}^{t-1}\left(\frac{\beta_*}{\theta}\right)^{t-1-s} \leq \theta^{t-1}\sum_{s=0}^{t-1}\theta^{t-1-s} \leq \theta^{t-1}\frac{4}{\sqrt{\eta\lambda}}$, (d) uses that $\theta \geq \frac{3}{4}$ and $\eta\lambda \leq 1$. Hence, we have shown (3.73). Therefore, by Theorem 24, we have $\left\|\begin{bmatrix}\xi_t\\ \xi_{t-1}\end{bmatrix}\right\| \leq \theta^t 2C_0\left\|\begin{bmatrix}\xi_0\\ \xi_{-1}\end{bmatrix}\right\|$.

By Lemma 29 and Lemma 31, we have (3.74). Furthermore, with the choice of m, we have $3n^2\exp(-mR^{\text{ReLU}}/10) \leq \delta$. Thus, we have completed the proof.

□

3.5.7 Proof of Theorem 27

We will need some supporting lemmas in the following for the proof. In the following analysis, we denote $C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H)),h(\beta,\eta\lambda_{\max}(H))\}}}$, where $h(\beta,\cdot)$ is the constant defined in Theorem 23 and $H = H_0 := \frac{1}{m^{L-1}d_y}\sum_{l=1}^{L}[(W_0^{(l-1:1)}X)^\top(W_0^{(l-1:1)}X)\otimes W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top] \in \mathbb{R}^{d_yn\times d_yn}$, as defined in Lemma 23. We also denote $\beta = (1-\frac{1}{2}\sqrt{\eta\lambda})^2 := \beta_*^2$. As mentioned in the main text, following Du and Hu [74], Hu, Xiao, and

Pennington [135], we will further assume that (A1) there exists a W^* such that $Y = W^*X$, $X \in \mathbb{R}^{d\times\bar{r}}$, and $\bar{r} = rank(X)$, which is actually without loss of generality (see e.g. the discussion in Section B of Du and Hu [74]). We summarize the notions in Table 3.2.

Notation	definition (or value)	meaning
$\mathcal{N}_W^{L\text{-linear}}(x)$	$\mathcal{N}_W^{L\text{-linear}}(x) := \frac{1}{\sqrt{m^{L-1}d_y}} W^{(L)}W^{(L-1)}\cdots W^{(1)}x,$	output of the deep linear network
H_0	$H_0 := \frac{1}{m^{L-1}d_y}\sum_{l=1}^{L}[(W_0^{(l-1:1)}X)^\top(W_0^{(l-1:1)}X) \otimes W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top] \in \mathbb{R}^{d_y n\times d_y n}$	H in (3.8) is $H = H_0$ (Lemma 23)
$\lambda_{\max}(H_0)$	$\lambda_{\max}(H_0) \le L\sigma^2_{\max}(X)/d_y$ (Lemma 32)	the largest eigenvalue of H_0
$\lambda_{\min}(H_0)$	$\lambda_{\min}(H_0) \ge L\sigma^2_{\min}(X)/d_y$ (Lemma 32)	the least eigenvalue of H_0
λ	$\lambda := L\sigma^2_{\min}(X)/d_y$	(a lower bound of) the least eigenvalue of H_0
κ	$\kappa := \frac{\lambda_1(X^\top X)}{\lambda_{\bar{r}}(X^\top X)} = \frac{\sigma^2_{\max}(X)}{\sigma^2_{\min}(X)}$ (A1)	the condition number of X
$\hat{\kappa}$	$\hat{\kappa} := \frac{\lambda_{\max}(H_0)}{\lambda_{\min}(H_0)} \le \frac{\sigma^2_{\max}(X)}{\sigma^2_{\min}(X)} = \kappa$ (Lemma 32)	the condition number of H_0
η	$\eta = \frac{d_y}{L\sigma^2_{\max}(X)}$	step size
β	$\beta = (1-\frac{1}{2}\sqrt{\eta\lambda})^2 = (1-\frac{1}{2\sqrt{\kappa}})^2 := \beta_*^2$	momentum parameter
β_*	$\beta_* = \sqrt{\beta} = 1-\frac{1}{2}\sqrt{\eta\lambda}$	squared root of β
θ	$\theta = \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1-\frac{1}{4}\sqrt{\eta\lambda} = 1-\frac{1}{4\sqrt{\kappa}}$	the convergence rate
C_0	$C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H_0)),h(\beta,\eta\lambda_{\max}(H_0))\}}}$	the constant used in Theorem 23

Table 3.2: Summary of the notations for proving Theorem 27. We will simply use κ to represent the condition number of the matrix H_0 in the analysis since we have $\hat{\kappa} \le \kappa$.

Lemma 32. *[Lemma 4.2 in Hu, Xiao, and Pennington [135]] By the orthogonal initialization, we have*

$$\lambda_{\min}(H_0) \ge L\sigma^2_{\min}(X)/d_y, \quad \lambda_{\max}(H_0) \le L\sigma^2_{\max}(X)/d_y.$$
$$\sigma_{\max}(W_0^{(j:i)}) = m^{\frac{j-i+1}{2}}, \quad \sigma_{\min}(W_0^{(j:i)}) = m^{\frac{j-i+1}{2}}$$

Furthermore, with probability $1-\delta$*,*

$$\ell(W_0) \le B_0^2 = O\left(1 + \frac{\log(\bar{r}/\delta)}{d_y} + \|W_*\|_2^2\right),$$

for some constant $B_0 > 0$*.*

We remark that Lemma 32 implies that the condition number of H_0 satisfies

$$\hat{\kappa} := \frac{\lambda_{\max}(H_0)}{\lambda_{\min}(H_0)} \leq \frac{\sigma_{\max}^2(X)}{\sigma_{\min}^2(X)} = \kappa. \tag{3.76}$$

Lemma 33. *Following the setting as Theorem 27, denote $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. If we have (I) for any $s \leq t$, the residual dynamics satisfies $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \leq \theta^s \cdot \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$, for some constant $\nu > 0$, and (II) for all $l \in [L]$ and for any $s \leq t$, $\|W_s^{(l)} - W_0^{(l)}\|_F \leq R^{\text{L-linear}} := \frac{64\|X\|_2\sqrt{d_y}}{L\sigma_{\min}^2(X)}\nu C_0 B_0$, then*

$$\|\phi_t\| \leq \frac{43\sqrt{d_y}}{\sqrt{m}\|X\|_2}\theta^{2t}\nu^2 C_0^2 \left(\frac{\|\xi_0\|}{1-\theta}\right)^2, \quad \|\psi_t\| \leq \frac{43\sqrt{d_y}}{\sqrt{m}\|X\|_2}\theta^{2(t-1)}\nu^2 C_0^2 \left(\frac{\|\xi_0\|}{1-\theta}\right)^2,$$
$$\|\iota_t\| \leq \frac{\eta\lambda}{80}\theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|.$$

Consequently, φ_t in Lemma 23 satisfies

$$\|\varphi_t\| \leq \frac{1920\sqrt{d_y}}{\sqrt{m}\|X\|_2}\frac{1}{\eta\lambda}\theta^{2t}\nu^2 C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|^2 + \frac{\eta\lambda}{80}\theta^t \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|.$$

Proof. By Lemma 23, $\varphi_t = \phi_t + \psi_t + \iota_t \in \mathbb{R}^{d_y n}$, we have

$$\phi_t := \frac{1}{\sqrt{m^{L-1}d_y}}\text{vec}(\Phi_t X)$$
$$\text{, with } \Phi_t := \Pi_l \left(W_t^{(l)} - \eta M_{t,l}\right) - W_t^{(L:1)} + \eta \sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W_t^{(l-1:1)}, \tag{3.77}$$

and

$$\psi_t := \frac{1}{\sqrt{m^{L-1}d_y}}\text{vec}\left((L-1)\beta W_t^{(L:1)}X + \beta W_{t-1}^{(L:1)}X - \beta \sum_{l=1}^{L} W_t^{(L:l+1)} W_{t-1}^{(l)} W_t^{(l-1:1)} X\right). \tag{3.78}$$

and

$$\iota_t := \eta(H_0 - H_t)\xi_t. \tag{3.79}$$

So if we can bound $\|\phi_t\|$, $\|\psi_t\|$, and $\|\iota_t\|$ respectively, then we can bound $\|\varphi_t\|$ by the triangle inequality.

$$\|\varphi_t\| \leq \|\phi_t\| + \|\psi_t\| + \|\iota_t\|. \tag{3.80}$$

Let us first upper-bound $\|\phi_t\|$. Note that Φ_t is the sum of all the high-order (of η's) term in the product,

$$W_{t+1}^{(L:1)} = \Pi_l \left(W_t^{(l)} - \eta M_{t,l}\right) = W_t^{(L:1)} - \eta \sum_{l=1}^{L} W_t^{(L:l+1)} M_{t,l} W^{(l-1:1)} + \Phi_t. \tag{3.81}$$

By induction, we can bound the gradient norm of each layer as

$$\begin{aligned}
\|\frac{\partial \ell(W_s^{(L:1)})}{\partial W_s^{(l)}}\|_F &\leq \frac{1}{\sqrt{m^{L-1}d_y}}\|W_s^{(L:l+1)}\|_2\|U_s - Y\|_F\|W_s^{(l-1:1)}\|_2\|X\|_2 \\
&\leq \frac{1}{\sqrt{m^{L-1}d_y}}1.1m^{\frac{L-l}{2}}\theta^s\nu C_0 2\sqrt{2}\|U_0 - Y\|_F 1.1m^{\frac{l-1}{2}}\|X\|_2 \\
&\leq \frac{4\|X\|_2}{\sqrt{d_y}}\theta^s\nu C_0\|U_0 - Y\|_F,
\end{aligned} \tag{3.82}$$

where the second inequality we use Lemma 35 and that $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \leq \theta^s \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$ and $\|\xi_s\| = \|U_s - Y\|_F$.

So the momentum term of each layer can be bounded as

$$\begin{aligned}\|M_{t,l}\|_F &= \|\sum_{s=0}^{t}\beta^{t-s}\frac{\partial \ell(W_s^{(L:1)})}{\partial W_s^{(l)}}\|_F \le \sum_{s=0}^{t}\beta^{t-s}\|\frac{\partial \ell(W_s^{(L:1)})}{\partial W_s^{(l)}}\|_F \\ &\le \frac{4\|X\|_2}{\sqrt{d_y}}\sum_{s=0}^{t}\beta^{t-s}\theta^s\nu C_0\|U_0-Y\|_F. \\ &\le \frac{4\|X\|_2}{\sqrt{d_y}}\sum_{s=0}^{t}\theta^{2(t-s)}\theta^s\nu C_0\|U_0-Y\|_F. \\ &\le \frac{4\|X\|_2}{\sqrt{d_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F,\end{aligned} \tag{3.83}$$

where in the second to last inequality we use $\beta = \beta_*^2 \le \theta^2$.

Combining all the pieces together, we can bound $\|\frac{1}{\sqrt{m^{L-1}d_y}}\Phi_t X\|_F$ as

$$\begin{aligned}&\|\frac{1}{\sqrt{m^{L-1}d_y}}\Phi_t X\|_F \\ &\overset{(a)}{\le} \frac{1}{\sqrt{m^{L-1}d_y}}\sum_{j=2}^{L}\binom{L}{j}\left(\eta\frac{4\|X\|_2}{\sqrt{d_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j(1.1)^{j+1}m^{\frac{L-j}{2}}\|X\|_2 \\ &\overset{(b)}{\le} 1.1\frac{1}{\sqrt{m^{L-1}d_y}}\sum_{j=2}^{L}L^j\left(\eta\frac{4.4\|X\|_2}{\sqrt{d_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j m^{\frac{L-j}{2}}\|X\|_2 \\ &\le 1.1\sqrt{\frac{m}{d_y}}\|X\|_2\sum_{j=2}^{L}\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j,\end{aligned} \tag{3.84}$$

where (a) uses (3.83) and Lemma 35 for bounding a $j \ge 2$ higher-order terms like

$$\frac{1}{\sqrt{m^{L-1}d_y}}\beta W_t^{(L:k_j+1)}\cdot(-\eta M_{t,k_j})W_t^{(k_j-1:k_{j-1}+1)}\cdot(-\eta M_{t,k_{j-1}})\cdots(-\eta M_{t,k_1})\cdot W_t^{(k_1-1:1)}$$

, where $1 \le k_1 < \cdots < k_j \le L$ and (b) uses that $\binom{L}{j} \le \frac{L^j}{j!}$

To proceed, let us bound $\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F$ in the sum above. We have

$$\begin{aligned}\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F &\leq 4.4\sqrt{\frac{d_y}{m}}\frac{1}{\|X\|_2}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F \\ &\leq 0.5,\end{aligned} \tag{3.85}$$

where the last inequality uses that $\tilde{C}_1\frac{d_yB_0^2C_0^2}{\|X\|_2^2}\frac{1}{(1-\theta)^2}\leq\tilde{C}_1\frac{d_yB_0^2C_0^2}{\|X\|_2^2}\frac{1}{\eta\lambda}\leq\tilde{C}_2\frac{d_yB_0^2\kappa^2}{\|X\|_2^2}\leq m$, for some sufficiently large constant $\tilde{C}_1,\tilde{C}_2>0$. Combining the above results, we have

$$\begin{aligned}\|\phi_t\| &= \|\frac{1}{\sqrt{m^{L-1}d_y}}\Phi_tX\|_F \\ &\leq 1.1\sqrt{\frac{m}{d_y}}\|X\|_2\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2\sum_{j=2}^{L-2}(0.5)^{j-2} \\ &\leq 2.2\sqrt{\frac{m}{d_y}}\|X\|_2\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2 \\ &\leq \frac{43\sqrt{d_y}}{\sqrt{m}\|X\|_2}\left(\frac{\theta^t}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2.\end{aligned} \tag{3.86}$$

Now let us switch to upper-bound $\|\psi_t\|$. It is equivalent to upper-bounding the Frobenius norm of $\frac{1}{\sqrt{m^{L-1}d_y}}\beta(L-1)W_t^{(L:1)}X+\frac{1}{\sqrt{m^{L-1}d_y}}\beta W_{t-1}^{(L:1)}X$
$-\frac{1}{\sqrt{m^{L-1}d_y}}\beta\sum_{l=1}^{L}W_t^{(L:l+1)}W_{t-1}^{(l)}W_t^{(l-1:1)}X$, which can be rewritten as

$$\begin{aligned}&\underbrace{\frac{1}{\sqrt{m^{L-1}d_y}}\beta(L-1)\cdot\Pi_{l=1}^{L}\left(W_{t-1}^{(l)}-\eta M_{t-1,l}\right)X}_{\text{first term}}+\underbrace{\frac{1}{\sqrt{m^{L-1}d_y}}\beta W_{t-1}^{(L:1)}X}_{\text{second term}} \\ &\underbrace{-\frac{1}{\sqrt{m^{L-1}d_y}}\beta\sum_{l=1}^{L}\Pi_{i=l+1}^{L}\left(W_{t-1}^{(i)}-\eta M_{t-1,i}\right)W_{t-1}^{(l)}\Pi_{j=1}^{l-1}\left(W_{t-1}^{(j)}-\eta M_{t-1,j}\right)X}_{\text{third term}}.\end{aligned} \tag{3.87}$$

The above can be written as $B_0+\eta B_1+\eta^2B_2+\cdots+\eta^LB_L$ for some matrices $B_0,\ldots,B_L\in$

$\mathbb{R}^{d_y \times n}$. Specifically, we have

$$
\begin{aligned}
B_0 &= \underbrace{\frac{1}{\sqrt{m^{L-1}d_y}}(L-1)\beta W_{t-1}^{(L:1)}X}_{\text{due to the first term}} + \underbrace{\frac{1}{\sqrt{m^{L-1}d_y}}\beta W_{t-1}^{(L:1)}X}_{\text{due to the second term}} \underbrace{-\frac{1}{\sqrt{m^{L-1}d_y}}\beta L W_{t-1}^{(L:1)}X}_{\text{due to the third term}} = 0 \\
B_1 &= \underbrace{-\frac{1}{\sqrt{m^{L-1}d_y}}(L-1)\beta \sum_{l=1}^{L} W_{t-1}^{(L:l+1)} M_{t-1,l} W_{t-1}^{(l-1:1)}}_{\text{due to the first term}} \\
&\quad + \underbrace{\frac{1}{\sqrt{m^{L-1}d_y}}\beta \sum_{l=1}^{L}\sum_{k\neq l} W_{t-1}^{(L:k+1)} M_{t-1,k} W_{t-1}^{(k-1:1)}}_{\text{due to the third term}} = 0.
\end{aligned}
\tag{3.88}
$$

So what remains on (3.87) are all the higher-order terms (in terms of the power of η), i.e. those with $\eta M_{t-1,i}$ and $\eta M_{t-1,j}$, $\forall i \neq j$ or higher.

To continue, observe that for a fixed (i,j), $i < j$, the second-order term that involves $\eta M_{t-1,i}$ and $\eta M_{t-1,j}$ on (3.87) is with coefficient $\frac{1}{\sqrt{m^{L-1}d_y}}\beta$, because the first term on (3.87) contributes to $\frac{1}{\sqrt{m^{L-1}d_y}}(L-1)\beta$, while the third term on (3.87) contributes to $-\frac{1}{\sqrt{m^{L-1}d_y}}(L-2)\beta$. Furthermore, for a fixed (i,j,k), $i<j<k$, the third-order term that involves $\eta M_{t-1,i}$, $\eta M_{t-1,j}$, and $\eta M_{t-1,k}$ on (3.87) is with coefficient $-2\frac{1}{\sqrt{m^{L-1}d_y}}\beta$, as the first term on (3.87) contributes to $-\frac{1}{\sqrt{m^{L-1}d_y}}(L-1)\beta$, while the third term on (3.87) contributes to $\frac{1}{\sqrt{m^{L-1}d_y}}(L-3)\beta$. Similarly, for a p-order term $\eta \underbrace{M_{t-1,*}, \cdots, \eta M_{t-1,**}}_{\text{p terms}}$, the coefficient is $(p-1)\frac{1}{\sqrt{m^{L-1}d_y}}\beta(-1)^p$.

By induction (see (3.83)), we can bound the norm of the momentum at layer l as

$$
\|M_{t-1,l}\|_F \leq \frac{4\|X\|_2}{\sqrt{d_y}} \frac{\theta^{t-1}}{1-\theta} \nu C_0 \|U_0 - Y\|_F. \tag{3.89}
$$

Combining all the pieces together, we have

$$
\begin{aligned}
&\|\frac{1}{\sqrt{m^{L-1}d_y}}\beta(L-1)W_t^{(L:1)}X + \frac{1}{\sqrt{m^{L-1}d_y}}\beta W_{t-1}^{(L:1)}X \\
&\qquad - \frac{1}{\sqrt{m^{L-1}d_y}}\beta\sum_{l=1}^{L} W_t^{(L:l+1)}W_{t-1}^{(l)}W_t^{(l-1:1)}X\|_F \\
&\overset{(a)}{\leq} \frac{\beta}{\sqrt{m^{L-1}d_y}}\sum_{j=2}^{L}(j-1)\binom{L}{j}\left(\eta\frac{4\|X\|_2}{\sqrt{d_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j(1.1)^{j+1}m^{\frac{L-j}{2}}\|X\|_2 \\
&\overset{(b)}{\leq} 1.1\frac{\beta}{\sqrt{m^{L-1}d_y}}\sum_{j=2}^{L}L^j\left(\eta\frac{4.4\|X\|_2}{\sqrt{d_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j m^{\frac{L-j}{2}}\|X\|_2 \\
&\leq 1.1\beta\sqrt{\frac{m}{d_y}}\|X\|_2\sum_{j=2}^{L}\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^j,
\end{aligned}
\tag{3.90}
$$

where (a) uses (3.89), the above analysis of the coefficients of the higher-order terms and Lemma 35 for bounding a $j \geq 2$ higher-order terms like $\frac{1}{\sqrt{m^{L-1}d_y}}\beta(j-1)(-1)^j W_{t-1}^{(L:k_j+1)} \cdot (-\eta M_{t-1,k_j})W_{t-1}^{(k_j-1:k_{j-1}+1)} \cdot (-\eta M_{t-1,k_{j-1}})\cdots(-\eta M_{t-1,k_1}) \cdot W_{t-1}^{(k_1-1:1)}$, where $1 \leq k_1 < \cdots < k_j \leq L$ and (b) uses that $\binom{L}{j} \leq \frac{L^j}{j!}$

Let us bound $\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F$ in the sum above. We have

$$
\begin{aligned}
\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F &\leq 4.4\sqrt{\frac{d_y}{m}}\frac{1}{\|X\|_2}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F \\
&\leq 0.5,
\end{aligned}
\tag{3.91}
$$

where the last inequality uses that $\tilde{C}_1\frac{d_yB_0^2C_0^2}{\|X\|_2^2}\frac{1}{(1-\theta)^2} \leq \tilde{C}_1\frac{d_yB_0^2C_0^2}{\|X\|_2^2}\frac{1}{\eta\lambda} \leq \tilde{C}_2\frac{d_yB_0^2\kappa^2}{\|X\|_2^2} \leq m$, for some sufficiently large constant $\tilde{C}_1, \tilde{C}_2 > 0$. Combining the above results, i.e. (3.90) and

(3.91), we have

$$
\begin{aligned}
\|\psi_t\| &\leq \|\frac{1}{\sqrt{m^{L-1}d_y}}\beta(L-1)W_t^{(L:1)}X + \frac{1}{\sqrt{m^{L-1}d_y}}\beta W_{t-1}^{(L:1)}X \\
&\qquad\qquad - \frac{1}{\sqrt{m^{L-1}d_y}}\beta\sum_{l=1}^{L} W_t^{(L:l+1)}W_{t-1}^{(l)}W_t^{(l-1:1)}X\|_F \\
&\leq 1.1\beta\sqrt{\frac{m}{d_y}}\|X\|_2\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2\sum_{j=2}^{L-2}(0.5)^{j-2} \qquad (3.92) \\
&\leq 2.2\beta\sqrt{\frac{m}{d_y}}\|X\|_2\left(\eta\frac{4.4L\|X\|_2}{\sqrt{md_y}}\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2 \\
&\leq \frac{43\sqrt{d_y}}{\sqrt{m}\|X\|_2}\left(\frac{\theta^{t-1}}{1-\theta}\nu C_0\|U_0-Y\|_F\right)^2,
\end{aligned}
$$

where the last inequality uses $\eta \leq \frac{d_y}{L\|X\|_2^2}$.

Now let us switch to bound $\|\iota_t\|$. We have

$$
\begin{aligned}
&\|\iota_t\| = \|\eta(H_t-H_0)\xi_t\| \\
&= \frac{\eta}{m^{L-1}d_y}\|\sum_{l=1}^{L} W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top(U_t-Y)(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X \\
&\qquad\qquad - \sum_{l=1}^{L} W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top(U_t-Y)(W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\|_F \\
&\leq \frac{\eta}{m^{L-1}d_y}\sum_{l=1}^{L}\|W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top(U_t-Y)(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X \\
&\qquad\qquad - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top(U_t-Y)(W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\|_F \\
&\leq \frac{\eta}{m^{L-1}d_y} \\
&\times\sum_{l=1}^{L}\Big(\underbrace{\|\left(W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\right)(U_t-Y)(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_F}_{\text{first term}} \\
&+\underbrace{\|W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top(U_t-Y)\left(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\right)\|_F\Big)}_{\text{second term}}.
\end{aligned}
\qquad (3.93)
$$

Now let us bound the first term. We have

$$\underbrace{\| \left(W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top \right) (U_t - Y)(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_F}_{\text{first term}}$$
$$\leq \|W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2 \|U_t - Y\|_F \|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_2. \tag{3.94}$$

For $\|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_2$, by using Lemma 34 and Lemma 35, we have

$$\|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_2 \leq \left(\sigma_{\max}(W_t^{(l-1:1)}X)\right)^2 \leq \left(1.1m^{\frac{l-1}{2}}\sigma_{\max}(X)\right)^2. \tag{3.95}$$

For $\|W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2$, denote $W_t^{(L:l+1)} = W_0^{(L:l+1)} + \Delta_t^{(L:l+1)}$, we have

$$\begin{aligned}
&\|W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2 \\
&\leq \|\Delta_t^{(L:l+1)}(W_t^{(L:l+1)})^\top + W_t^{(L:l+1)}(\Delta_t^{(L:l+1)})^\top + \Delta_t^{(L:l+1)}(\Delta_t^{(L:l+1)})^\top\|_2 \\
&\leq 2\|\Delta_t^{(L:l+1)}\|_2 \cdot \sigma_{\max}(W_t^{(L:l+1)}) + \|\Delta_t^{(L:l+1)}\|_2^2 \\
&\leq 2\|\Delta_t^{(L:l+1)}\|_2 \cdot \left(1.1m^{\frac{L-l}{2}}\right) + \|\Delta_t^{(L:l+1)}\|_2^2.
\end{aligned} \tag{3.96}$$

Therefore, we have to bound $\|\Delta_t^{(L:l+1)}\|_2$. We have for any $1 \leq i \leq j \leq L$.

$$W_t^{(j:i)} = \left(W_0^{(j)} + \Delta_j\right) \cdots \left(W_0^{(i)} + \Delta_i\right), \tag{3.97}$$

where $\|\Delta_i\|_2 \leq \|W_t^{(i)} - W_0^{(i)}\|_F \leq D := \frac{64\|X\|_2\sqrt{d_y}}{L\sigma_{\min}^2(X)}\nu C_0 B_0$ by Lemma 34. The product (3.97) above minus $W_0^{(j:i)}$ can be written as a finite sum of some terms of the form

$$W_0^{(j:k_l+1)}\Delta_{k_l}W_0^{(k_l-1:k_{l-1}+1)}\Delta_{k_{l-1}}\cdots\Delta_{k_1}W_0^{(k_1-1:i)}, \tag{3.98}$$

where $i \leq k_1 < \cdots < k_l \leq j$. Recall that $\|W_0^{(j':i')}\|_2 = m^{\frac{j'-i'+1}{2}}$ by Lemma 32. Thus, we can bound

$$
\begin{aligned}
&\|\Delta_t^{(j:i)}\|_2 \le \|W_t^{(j:i)} - W_0^{(j:i)}\|_F \\
&\le \sum_{l=1}^{j-i+1} \binom{j-i+1}{l} (D)^l m^{\frac{j-i+1-l}{2}} = (\sqrt{m}+D)^{j-i+1} - (\sqrt{m})^{j-i+1} \\
&= (\sqrt{m})^{j-i+1}\left((1+D/\sqrt{m})^{j-i+1} - 1\right) \le (\sqrt{m})^{j-i+1}\left((1+D/\sqrt{m})^{L} - 1\right) \qquad (3.99) \\
&\overset{(a)}{=} \left((1+\frac{1}{\sqrt{C'}L\kappa})^L - 1\right)(\sqrt{m})^{j-i+1} \overset{(b)}{\le} \left(\exp\left(\frac{1}{\sqrt{C'}\kappa}\right) - 1\right)(\sqrt{m})^{j-i+1} \\
&\overset{(c)}{\le} \left(1 + (e-1)\frac{1}{\sqrt{C'}\kappa} - 1\right)(\sqrt{m})^{j-i+1} \overset{(d)}{\le} \frac{1}{480\kappa}(\sqrt{m})^{j-i+1},
\end{aligned}
$$

where (a) uses $\frac{D}{\sqrt{m}} \le \frac{1}{\sqrt{C'}L\kappa}$, for some constant $C' > 0$, since $C'\frac{d_y C_0^2 B_0^2 \kappa^4}{\|X\|_2^2} \le C\frac{d_y B_0^2 \kappa^5}{\|X\|_2^2} \le m$, (b) follows by the inequality $(1+x/n)^n \le e^x, \forall x \ge 0, n > 0$, (c) from Bernoulli's inequality $e^r \le 1 + (e-1)r, \forall 0 \le r \le 1$, and (d) by choosing any sufficiently larger C'.

From (3.99), we have $\|\Delta_t^{(L:l+1)}\|_2 \le \frac{1}{480\kappa}(\sqrt{m})^{L-l}$. Combining this with (3.94), (3.95), and (3.96), we have

$$
\begin{aligned}
&\underbrace{\|\left(W_t^{(L:l+1)}(W_t^{(L:l+1)})^\top - W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\right)(U_t - Y)(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_F}_{\text{first term}} \\
&\le \left(2\|\Delta_t^{(L:l+1)}\|_2 \cdot \left(1.1 m^{\frac{L-l}{2}}\right) + \|\Delta_t^{(L:l+1)}\|_2^2\right)\left(1.1 m^{\frac{l-1}{2}}\sigma_{\max}(X)\right)^2 \|U_t - Y\|_F \\
&\le \left(2\frac{1}{480\kappa}(\sqrt{m})^{L-l}\cdot\left(1.1 m^{\frac{L-l}{2}}\right) + \left(\frac{1}{480\kappa}(\sqrt{m})^{L-l}\right)^2\right)\left(1.1 m^{\frac{l-1}{2}}\sigma_{\max}(X)\right)^2 \|U_t - Y\|_F \\
&\le \frac{\sigma_{\min}^2(X)}{160} m^{L-1}\|U_t - Y\|_F,
\end{aligned}
\qquad (3.100)
$$

where in the last inequality we use $\kappa := \frac{\sigma_{\max}^2(X)}{\sigma_{\min}^2(X)}$.

Now let us switch to bound the second term, we have

$$\underbrace{\|(W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top(U_t - Y)\left(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\right)\|_F)}_{\text{second term}}$$

$$\leq \|(W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2\|U_t - Y\|_F\|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\|_2. \tag{3.101}$$

For $\|W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2$, based on Lemma 32, we have

$$\|W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top\|_2 \leq m^{L-l}. \tag{3.102}$$

To bound $\|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\|_2$, we proceed as follows. Denote $W_t^{(l-1:1)} = W_0^{(l-1:1)} + \Delta_t^{(l-1:1)}$, we have

$$\begin{aligned}
&\|(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\|_2 \\
&\leq 2\|(\Delta_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X\|_2 + \|\Delta_t^{(l-1:1)}X\|_2^2 \\
&\leq \left(2\,\|\Delta_t^{(l-1:1)}\|\|W_t^{(l-1:1)}\|_2 + \|\Delta_t^{(l-1:1)}\|_2^2\right)\|X\|_2^2 \\
&\leq \left(2\frac{1}{480\kappa}m^{\frac{l-1}{2}}1.1m^{\frac{l-1}{2}} + \left(\frac{1}{480\kappa}m^{\frac{l-1}{2}}\right)^2\right)\|X\|_2^2 \\
&\leq \frac{\sigma_{\min}^2(X)}{160}m^{l-1},
\end{aligned} \tag{3.103}$$

where the second to last inequality uses (3.99), Lemma 34, and Lemma 35, while the last inequality uses $\kappa := \frac{\sigma_{\max}^2(X)}{\sigma_{\min}^2(X)}$. Combining (3.101), (3.102), (3.103), we have

$$\underbrace{\|(W_0^{(L:l+1)}(W_0^{(L:l+1)})^\top(U_t - Y)\left(W_t^{(l-1:1)}X)^\top W_t^{(l-1:1)}X - (W_0^{(l-1:1)}X)^\top W_0^{(l-1:1)}X\right)\|_F)}_{\text{second term}}$$

$$\leq \frac{\sigma_{\min}^2(X)}{160}m^{L-1}\|U_t - Y\|_F. \tag{3.104}$$

Now combing (3.93), (3.100), and (3.104), we have

$$\|\iota_t\| \leq \frac{\eta}{m^{L-1} d_y} L \frac{\sigma^2_{\min}(X)}{80} m^{L-1} \|U_t - Y\|_F = \frac{\eta\lambda}{80} \|\xi_t\|, \tag{3.105}$$

where we use $\lambda := \frac{L\sigma^2_{\min}(X)}{d_y}$.

Now we have (3.86), (3.92), and (3.105), which leads to

$$\begin{aligned} \|\varphi_t\| &\leq \|\phi_t\| + \|\psi_t\| + \|\iota_t\| \\ &\leq \frac{43\sqrt{d_y}}{\sqrt{m}\|X\|_2} (\theta^{2t} + \theta^{2(t-1)}) \nu^2 C_0^2 \left(\frac{\|\xi_0\|}{1-\theta} \right)^2 + \frac{\eta\lambda}{80} \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|. \\ &\leq \frac{1920\sqrt{d_y}}{\sqrt{m}\|X\|_2} \frac{1}{\eta\lambda} \theta^{2t} \nu^2 C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|^2 + \frac{\eta\lambda}{80} \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|. \end{aligned} \tag{3.106}$$

where the last inequality uses that $1 \leq \frac{16}{9}\theta^2$ as $\eta\lambda \leq 1$ so that $\theta \geq \frac{3}{4}$. $\square$

Lemma 34. *Following the setting as Theorem 27, denote $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. If for any $s \leq t$, the residual dynamics satisfies $\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \| \leq \theta^s \cdot \nu C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$, for some constant $\nu > 0$, then*

$$\|W_t^{(l)} - W_0^{(l)}\|_F \leq R^{\text{L-linear}} := \frac{64\|X\|_2\sqrt{d_y}}{L\sigma^2_{\min}(X)} \nu C_0 B_0.$$

Proof. We have

$$
\begin{aligned}
\|W_{t+1}^{(l)} - W_0^{(l)}\|_F &\overset{(a)}{\leq} \eta \sum_{s=0}^{t} \|M_{s,l}\|_F \overset{(b)}{=} \eta \sum_{s=0}^{t} \| \sum_{\tau=0}^{s} \beta^{s-\tau} \frac{\partial \ell(W_\tau^{(L:1)})}{\partial W_\tau^{(L)}} \|_F \\
&\leq \eta \sum_{s=0}^{t} \sum_{\tau=0}^{s} \beta^{s-\tau} \| \frac{\partial \ell(W_\tau^{(L:1)})}{\partial W_\tau^{(L)}} \|_F \\
&\overset{(c)}{\leq} \eta \sum_{s=0}^{t} \sum_{\tau=0}^{s} \beta_*^{2(s-\tau)} \frac{4\|X\|_2}{\sqrt{d_y}} \theta^\tau \nu C_0 \|U_0 - Y\|_F. \\
&\overset{(d)}{\leq} \eta \sum_{s=0}^{t} \frac{\theta^s}{1-\theta} \frac{4\|X\|_2}{\sqrt{d_y}} \nu C_0 \|U_0 - Y\|_F. \\
&\leq \frac{4\eta\|X\|_2}{\sqrt{d_y}} \frac{1}{(1-\theta)(1-\theta)} \nu C_0 \|U_0 - Y\|_F \\
&\overset{(e)}{\leq} \frac{64\|X\|_2}{\lambda\sqrt{d_y}} \nu C_0 \|U_0 - Y\|_F \\
&\overset{(f)}{\leq} \frac{64\|X\|_2\sqrt{d_y}}{L\sigma_{\min}^2(X)} \nu C_0 B_0,
\end{aligned}
\tag{3.107}
$$

where (a), (b) is by the update rule of momentum, which is $W_{t+1}^{(l)} - W_t^{(l)} = -\eta M_{t,l}$, where $M_{t,l} := \sum_{s=0}^{t} \beta^{t-s} \frac{\partial \ell(W_{L:1})}{\partial W_s^{(l)}}$, (c) is because $\| \frac{\partial \ell(W_{L:1})}{\partial W_s^{(l)}} \|_F = \frac{4\|X\|_2}{\sqrt{d_y}} \theta^s \nu C_0 \|U_0 - Y\|_F$ (see (3.82)), (d) is because that $\beta = \beta_*^2 \leq \theta^2$, (e) is because that $\frac{1}{(1-\theta)^2} = \frac{16}{\eta\lambda}$, and (f) uses the upper-bound $B_0 \geq \|U_0 - Y\|$ defined in Lemma 32 and $\lambda := \frac{L\sigma_{\min}^2(X)}{d_y}$. The proof is completed.

∎

Lemma 35. *Hu, Xiao, and Pennington [135] Let $R^{\text{L-linear}}$ be an upper bound that satisfies $\|W_t^{(l)} - W_t^{(l)}\|_F \leq R^{\text{L-linear}}$ for all l and t. Suppose the width m satisfies $m > C(LR^{\text{L-linear}})^2$, where C is any sufficiently large constant. Then,*

$$
\sigma_{\max}(W_t^{(j:i)}) \leq 1.1 m^{\frac{j-i+1}{2}}, \sigma_{\min}(W_t^{(j:i)}) \geq 0.9 m^{\frac{j-i+1}{2}}.
$$

Proof. The lemma has been proved in proof of Claim 4.4 and Claim 4.5 in Hu, Xiao, and Pennington [135]. For completeness, let us replicate the proof here.

We have for any $1 \leq i \leq j \leq L$.

$$W_t^{(j:i)} = \left(W_0^{(j)} + \Delta_j\right) \cdots \left(W_0^{(i)} + \Delta_i\right), \tag{3.108}$$

where $\Delta_i = W_t^{(i)} - W_0^{(i)}$. The product above minus $W_0^{(j:i)}$ can be written as a finite sum of some terms of the form

$$W_0^{(j:k_l+1)} \Delta_{k_l} W_0^{(k_l-1:k_{l-1}+1)} \Delta_{k_{l-1}} \cdots \Delta_{k_1} W_0^{(k_1-1:i)}, \tag{3.109}$$

where $i \leq k_1 < \cdots < k_l \leq j$. Recall that $\|W_0^{(j':i')}\|_2 = m^{\frac{j'-i'+1}{2}}$. Thus, we can bound

$$\begin{aligned}
\|W_t^{(j:i)} - W_0^{(j:i)}\|_F &\leq \sum_{l=1}^{j-i+1} \binom{j-i+1}{l} (R^{L\text{-linear}})^l m^{\frac{j-i+1-l}{2}} \\
&= (\sqrt{m} + R^{L\text{-linear}})^{j-i+1} - (\sqrt{m})^{j-i+1} \\
&= (\sqrt{m})^{j-i+1} \left((1 + R^{L\text{-linear}}/\sqrt{m})^{j-i+1} - 1\right) \\
&\leq (\sqrt{m})^{j-i+1} \left((1 + R^{L\text{-linear}}/\sqrt{m})^{L} - 1\right) \\
&\leq 0.1(\sqrt{m})^{j-i+1},
\end{aligned} \tag{3.110}$$

where the last step uses $m > C(LR^{L\text{-linear}})^2$. By combining this with Lemma 32, one can obtain the result.

$\square$

Remark: In the proof of Lemma 33, we obtain a tighter bound of the distance $\|W_t^{(j:i)} - W_0^{(j:i)}\|_F \leq O(\frac{1}{\kappa}(\sqrt{m})^{j-i+1})$. However, to get the upper-bound $\sigma_{\max}(W_t^{(j:i)})$ shown in Lemma 35, (3.110) is sufficient for the purpose.

3.5.8 Proof of Theorem 27

Proof. (of Theorem 27) Denote $\lambda := L\sigma_{\min}^2(X)/d_y$. By Lemma 32, $\lambda_{\min}(H) \geq \lambda$. Also, denote $\beta_* := 1 - \frac{1}{2}\sqrt{\eta\lambda}$ and $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. Let $\nu = 2$ in Lemma 33, 34, and let $C_1 = C_3 = C_0$ and $C_2 = \frac{1}{4}\sqrt{\eta\lambda}$ in Theorem 24. The goal is to show that $\left\|\begin{bmatrix}\xi_t \\ \xi_{t-1}\end{bmatrix}\right\| \leq \theta^t 2C_0 \left\|\begin{bmatrix}\xi_0 \\ \xi_{-1}\end{bmatrix}\right\|$ for all t by induction. To achieve this, we will also use induction to show that for all iterations s,

$$\forall l \in [L], \|W_t^{(l)} - W_0^{(l)}\| \leq R^{L\text{-linear}} := \frac{64\|X\|_2\sqrt{d_y}}{L\sigma_{\min}^2(X)} C_0 B_0, \tag{3.111}$$

which is clearly true in the base case $s = 0$.

By Lemma 23, 32, 33, 34, Theorem 24 and Corollary 3, it suffices to show that $\left\|\begin{bmatrix}\xi_s \\ \xi_{s-1}\end{bmatrix}\right\| \leq \theta^s \cdot 2C_0 \left\|\begin{bmatrix}\xi_0 \\ \xi_{-1}\end{bmatrix}\right\|$ and $\forall l \in [L], \|W_s^{(l)} - W_0^{(l)}\| \leq R^{L\text{-linear}}$ hold at $s = 0, 1, \ldots, t-1$, one has

$$\|\sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix}\varphi_s \\ 0\end{bmatrix}\| \leq \theta^t C_0 \left\|\begin{bmatrix}\xi_0 \\ \xi_{-1}\end{bmatrix}\right\|, \tag{3.112}$$

$$\forall l \in [L], \|W_t^{(l)} - W_0^{(l)}\| \leq R^{L\text{-linear}} := \frac{64\|X\|_2\sqrt{d_y}}{L\sigma_{\min}^2(X)} C_0 B_0, \tag{3.113}$$

where the matrix A and the vector φ_t are defined in Lemma 23, and B_0 is a constant such that $B_0 \geq \|Y - U_0\|_F$ with probability $1 - \delta$ by Lemma 32. The inequality (3.112) is the required condition for using the result of Theorem 24, while the inequality (3.113) helps us to show (3.112) through invoking Lemma 33 to bound the terms $\{\varphi_s\}$ as shown in the following.

Let us show (3.112) first. We have

$$
\begin{aligned}
\| \sum_{s=0}^{t-1} A^{t-1-s} \| \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix} \| &\overset{(a)}{\leq} \sum_{s=0}^{t-1} \beta_*^{t-1-s} C_0 \|\varphi_s\| \\
&\overset{(b)}{\leq} \frac{1920\sqrt{d_y}}{\sqrt{m}\|X\|_2} \frac{1}{\eta\lambda} \sum_{s=0}^{t-1} \beta_*^{t-1-s} \theta^{2s} 4C_0^3 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|^2 \\
&\quad + \sum_{s=0}^{t-1} \beta_*^{t-1-s} \frac{\eta\lambda}{80} \theta^s 2C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(c)}{\leq} \frac{1920\sqrt{d_y}}{\sqrt{m}\|X\|_2} \frac{1}{\eta\lambda} \sum_{s=0}^{t-1} \beta_*^{t-1-s} \theta^{2s} 4C_0^3 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|^2 \\
&\quad + \frac{2\sqrt{\eta\lambda}}{15} \theta^t C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(d)}{\leq} \frac{1920\sqrt{d_y}}{\sqrt{m}\|X\|_2} \frac{16}{3(\eta\lambda)^{3/2}} \theta^t 4C_0^3 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|^2 + \frac{2\sqrt{\eta\lambda}}{15} \theta^t C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(e)}{\leq} \frac{1}{3} \theta^t C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| + \frac{2\sqrt{\eta\lambda}}{15} \theta^t C_0^2 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(f)}{\leq} \theta^t C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|,
\end{aligned}
\tag{3.114}
$$

where (a) uses Theorem 23 with $\beta = \beta_*^2$, (b) is by Lemma 33, (c) uses $\sum_{s=0}^{t-1} \beta_*^{t-1-s}\theta^s = \theta^{t-1}\sum_{s=0}^{t-1}\left(\frac{\beta_*}{\theta}\right)^{t-1-s} \leq \theta^{t-1}\sum_{s=0}^{t-1}\theta^{t-1-s} \leq \theta^{t-1}\frac{4}{\sqrt{\eta\lambda}} \leq \theta^t\frac{16}{3\sqrt{\eta\lambda}}$, $\beta_* = 1 - \frac{1}{2}\sqrt{\eta\lambda} \geq \frac{1}{2}$, and $\theta = 1 - \frac{1}{4}\sqrt{\eta\lambda} \geq \frac{3}{4}$, (d) uses $\sum_{s=0}^{t-1}\beta_*^{t-1-s}\theta^{2s} \leq \sum_{s=0}^{t-1}\theta^{t-1+s} \leq \frac{\theta^{t-1}}{1-\theta} \leq \theta^t\frac{16}{3\sqrt{\eta\lambda}}$, (e) is because $C'\frac{d_y C_0^4 B_0^2}{\|X\|_2^2}\frac{1}{(\eta\lambda)^3} \leq C\frac{d_y B_0^2}{\|X\|_2^2}\kappa^5 \leq m$ for some sufficiently large constants $C', C > 0$, and (f) uses that $\eta\lambda = \frac{1}{\kappa}$ and $C_0 \leq 4\sqrt{\kappa}$ by Corollary 3. Hence, we have shown (3.112). Therefore, by Theorem 24, we have $\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \theta^t 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$.

By Lemma 34, we have (3.113). Thus, we have completed the proof.

$\square$

3.5.9 Non-asymptotic accelerated linear rate of the local convergence for solving $f(\cdot) \in F^2_{\mu,\alpha}$

Theorem 28. *Assume that the function $f(\cdot) \in F^2_{\mu,\alpha}$ and its Hessian is α-Lipschitz. Denote the condition number $\kappa := \frac{\alpha}{\mu}$. Suppose that the initial point satisfies $\| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \| \leq \frac{1}{683\kappa^{3/2}}$. Then, Gradient descent with Polyak's momentum with the step size $\eta = \frac{1}{\alpha}$ and the momentum parameter $\beta = \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^2$ for solving $\min_w f(w)$ has*

$$\| \begin{bmatrix} w_{t+1} - w_* \\ w_t - w_* \end{bmatrix} \| \leq \left(1 - \frac{1}{4\sqrt{\kappa}}\right)^{t+1} 8\sqrt{\kappa} \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \|, \tag{3.115}$$

where $w_* = \arg\min_w f(w)$.

Remark: Compared to Theorem 9 of Polyak [223], Theorem 28 clearly indicates the required distance that ensures an acceleration when the iterate is in the neighborhood of the global minimizer. Furthermore, the rate is in the non-asymptotic sense instead of the asymptotic one.

Proof. In the following, we denote $\xi_t := w_t - w_*$ and denote $\lambda := \mu > 0$, which is a lower bound of $\lambda_{\min}(H)$ of the matrix $H := \int_0^1 \nabla^2 f\big((1-\tau)w_0 + w_*\big) d\tau$ defined in Lemma 21, i.e. $\lambda_{\min}(H) \geq \lambda$. Also, denote $\beta_* := 1 - \frac{1}{2}\sqrt{\eta\lambda}$ and $\theta := \beta_* + \frac{1}{4}\sqrt{\eta\lambda} = 1 - \frac{1}{4}\sqrt{\eta\lambda}$. Suppose $\eta = \frac{1}{\alpha}$, where α is the smoothness constant. Denote $C_0 := \frac{\sqrt{2}(\beta+1)}{\sqrt{\min\{h(\beta,\eta\lambda_{\min}(H)), h(\beta,\eta\lambda_{\max}(H))\}}} \leq 4\sqrt{\kappa}$ by Corollary 3. Let $C_1 = C_3 = C_0$ and $C_2 = \frac{1}{4}\sqrt{\eta\lambda}$ in Theorem 24. The goal is to show that $\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \theta^t 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$ for all t by induction. To achieve this, we will also use induction to show that for all iterations s,

$$\|w_s - w_*\| \leq R := \tfrac{3}{64\sqrt{\kappa}C_0}. \tag{3.116}$$

A sufficient condition for the base case $s = 0$ of (3.116) to hold is

$$\left\| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \right\| \leq \frac{R}{2C_0} = \frac{3}{128\sqrt{\kappa}C_0^2}, \tag{3.117}$$

as $C_0 \geq 1$ by Theorem 23, which in turn can be guaranteed if $\left\| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \right\| \leq \frac{1}{683\kappa^{3/2}}$ by using the upper bound $C_0 \leq 4\sqrt{\kappa}$ of Corollary 3.

From Lemma 21, we have

$$\begin{aligned} \|\phi_s\| &\leq \eta \left\| \int_0^1 \nabla^2 f((1-\tau)w_s + \tau w_*) d\tau - \int_0^1 \nabla^2 f((1-\tau)w_0 + \tau w_*) d\tau \right\| \|\xi_s\| \\ &\overset{(a)}{\leq} \eta\alpha \left(\int_0^1 (1-\tau)\|w_s - w_0\| d\tau \right) \|\xi_s\| \leq \eta\alpha \|w_s - w_0\| \|\xi_s\| \\ &\overset{(b)}{\leq} \eta\alpha \left(\|w_s - w_*\| + \|w_0 - w_*\| \right) \|\xi_s\|, \end{aligned} \tag{3.118}$$

where (a) is by α-Lipschitzness of the Hessian and (b) is by the triangle inequality. By (3.116), (3.118), Lemma 21, Theorem 24, and Corollary 3, it suffices to show that given $\left\| \begin{bmatrix} \xi_s \\ \xi_{s-1} \end{bmatrix} \right\| \leq \theta^s 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$ and $\|w_s - w_*\| \leq R := \frac{3}{64\sqrt{\kappa}C_0}$ hold at $s = 0, 1, \ldots, t-1$, one has

$$\begin{aligned} \left\| \sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix} \right\| &\leq \theta^t C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\| && (3.119) \\ \|w_t - w_*\| &\leq R := \frac{3}{64\sqrt{\kappa}C_0}, && (3.120) \end{aligned}$$

where $A := \begin{bmatrix} (1+\beta)I_n - \eta \int_0^1 \nabla^2 f\big((1-\tau)w_0 + w_*\big) d\tau & -\beta I_n \\ I_n & 0 \end{bmatrix}$.

We have

$$
\begin{aligned}
\|\sum_{s=0}^{t-1} A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix} \| &\leq \sum_{s=0}^{t-1} \|A^{t-s-1} \begin{bmatrix} \varphi_s \\ 0 \end{bmatrix} \| \\
&\overset{(a)}{\leq} \sum_{s=0}^{t-1} \beta_*^{t-s-1} C_0 \|\varphi_s\| \\
&\overset{(b)}{\leq} 4\eta\alpha R C_0^2 \sum_{s=0}^{t-1} \beta_*^{t-s-1} \theta^s \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(c)}{\leq} R C_0^2 \frac{64}{3\sqrt{\eta\lambda}} \theta^t \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \| \\
&\overset{(d)}{\leq} C_0 \theta^t \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|,
\end{aligned}
\tag{3.121}
$$

where (a) uses Theorem 23 with $\beta = \beta_*^2$, (b) is by (3.118), (3.116), and the induction that $\|\xi_s\| \leq \theta^s 2C_0 \| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \|$, (c) is because $\sum_{s=0}^{t-1} \beta_*^{t-1-s}\theta^s = \theta^{t-1} \sum_{s=0}^{t-1} \left(\frac{\beta_*}{\theta}\right)^{t-1-s} \leq \theta^{t-1} \sum_{s=0}^{t-1} \theta^{t-1-s} \leq \theta^{t-1} \frac{4}{\sqrt{\eta\lambda}} \leq \theta^t \frac{16}{3\sqrt{\eta\lambda}}$, and (d) is due to the definition of $R := \frac{3}{64\sqrt{\kappa}C_0}$. Therefore, by Theorem 24, we have $\left\| \begin{bmatrix} \xi_t \\ \xi_{t-1} \end{bmatrix} \right\| \leq \theta^t 2C_0 \left\| \begin{bmatrix} \xi_0 \\ \xi_{-1} \end{bmatrix} \right\|$.

Now let us switch to show (3.120). We have

$$
\|\xi_t\| := \|w_t - w_*\| \overset{\text{induction}}{\leq} \theta^t 2C_0 \| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \| \leq R, \tag{3.122}
$$

where the last inequality uses the constraint $\| \begin{bmatrix} w_0 - w_* \\ w_{-1} - w_* \end{bmatrix} \| \leq \frac{R}{2C_0}$ by (3.117). $\square$

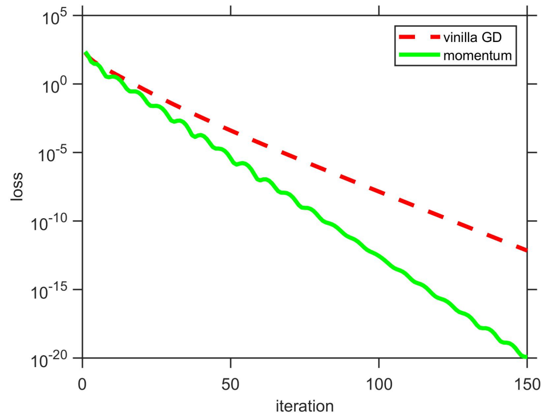


Figure 3.2: Training a 100-layer deep linear network. Here "momentum" stands for gradient descent with Polyak's momentum.

3.6 Experiments

3.6.1 ReLU network

We report a proof-of-concept experiment for training the ReLU network. We sample $n = 5$ points from the normal distribution, and then scale the size to the unit norm. We generate the labels uniformly random from $\{1, -1\}$. We let $m = 1000$ and $d = 10$. We compare vanilla GD and gradient descent with Polyak's momentum. We use the empirical Gram matrix at the initialization as an estimate of $\bar{H}$. Denote $\hat{\lambda}_{\max} := \lambda_{\max}(H_0)$, $\hat{\lambda}_{\min} := \lambda_{\min}(H_0)$, and $\hat{\kappa} := \hat{\lambda}_{\max}/\hat{\lambda}_{\min}$. Then, for gradient descent with Polyak's momentum, we set the step size $\eta = 1/\left(\hat{\lambda}_{\max}\right)$ and set the momentum parameter $\beta = (1 - \frac{1}{2}\frac{1}{\sqrt{\hat{\kappa}}})^2$. For gradient descent, we set the same step size. The result is shown on Figure 3.3.

We also report the percentiles of pattern changes over iterations. Specifically, we report the quantity

$$\frac{\sum_{i=1}^{n}\sum_{r=1}^{m}\mathbb{1}\{\text{sign}(x_i^\top w_t^{(r)}) \neq \text{sign}(x_i^\top w_0^{(r)})\}}{mn},$$

as there are mn patterns. For gradient descent with Polyak's momentum, the percentiles of pattern changes is approximately 0.76%; while for vanilla gradient descent, the percentiles of pattern changes is 0.55%.

3.6.2 Deep linear network

We let the input and output dimension $d = d_y = 20$, the width of the intermediate layers $m = 50$, the depth $L = 100$. We sampled a $X \in \mathbb{R}^{20\times 5}$ from the normal distribution. We let $W^* = I_{20} + 0.1\bar{W}$, where $\bar{W} \in \mathbb{R}^{20\times 20}$ is sampled from the normal distribution. Then, we have $Y = W^* X$, $\eta = \frac{d_y}{2L\sigma^2_{\max}(X)}$ and $\beta = (1 - \frac{1}{2}\sqrt{\eta\lambda})^2$, where $\lambda = \frac{L\sigma^2_{\min}(X)}{d_y}$. Vanilla GD also uses the same step size. The network is initialized by the orthogonal initialization and both algorithms start from the same initialization. The result is shown on Figure 3.2.

3.7 Conclusion

We show some non-asymptotic acceleration results of the discrete-time Polyak's momentum in this work. The results not only improve the previous results in convex optimization but also establish the first time that Polyak's momentum has provable acceleration for training certain neural networks. We analyze all the acceleration results from a modular framework. We hope the framework can serve as a building block towards understanding Polyak's momentum in a more unified way.

One of the possible future work is considering applying Polyak's momentum to the Nesterov-Polyak cubic-regularized problem [212],

$$\min_w f(w) := \tfrac{1}{2} w^\top A w + b^\top w + \tfrac{\rho}{3}\|w\|^3, \tag{3.123}$$

where the matrix $A \in \mathbb{R}^{d\times d}$ is symmetric and possibly indefinite. At the first glance, it looks a bit like the quadratic problems. However, due to the presence of the cubic-regularized term, the Hessian is changing and can change significantly during the optimization process. Empirically (Figure 3.3) we observe that Polyak's momentum leads to acceleration. Yet, as far as we know, no theoretical result in the literature is able to explain this observation. Therefore, it is interesting to check if our modular analysis can be extended to this problem as well.

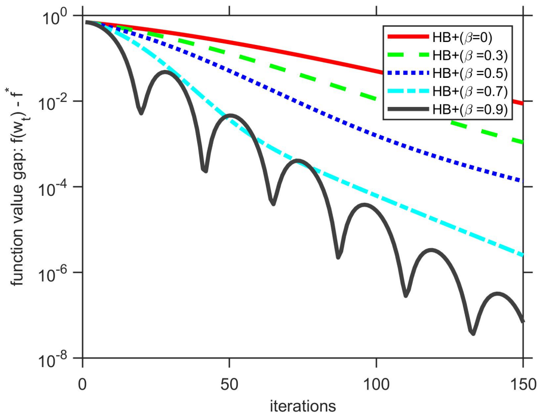


Figure 3.3: Optimality gap $f(w_t) - \min f(w)$ vs. iteration t. We see that a larger momentum leads to an acceleration. The setup of the experiment is as follows. We first set step size $\eta = 0.01$, dimension $d = 4$, $\rho = \|w_*\| = \|A\|_2 = 1$, $\lambda_{\min}(A) = -0.2$ and $\mathbf{gap} = 5 \times 10^{-3}$. Then we set $A = \mathrm{diag}([\lambda_{\min}(A); \lambda_{\min}(A) + \mathbf{gap}; a_{33}; a_{44}])$, where the entries a_{33} and a_{44} are sampled uniformly random in $[\lambda_{\min}(A) + \mathbf{gap}; \|A\|_2]$. We draw $\tilde{w} = (A + \rho\|w_*\|I_d)^{-\xi}\theta$, where $\theta \sim \mathcal{N}(0; I_d)$ and $\log_2 \xi$ is uniform on $[-1, 1]$. We set $w_* = \frac{\|w_*\|}{\|\tilde{w}\|}\tilde{w}$ and $b = -(A + \rho\|w_*\|I_d)w_*$. The procedure makes w_* the global minimizer of problem instance (A, b, ρ). Patterns shown on this figure exhibit for other random problem instances as well.

CHAPTER 4
ESCAPING SADDLE POINTS FASTER VIA POLYAK'S MOMENTUM

4.1 Introduction

SGD with stochastic momentum has been a de facto algorithm in nonconvex optimization and deep learning. It has been widely adopted for training machine learning models in various applications. Modern techniques in computer vision (e.g.[156, 129, 60, 104]), speech recognition (e.g. [12]), natural language processing (e.g. [263]), and reinforcement learning (e.g. [242]) use SGD with stochastic momentum to train models. The advantage of SGD with stochastic momentum has been widely observed ([131, 185]). Sutskever et al. [252] demonstrate that training deep neural nets by SGD with stochastic momentum helps achieving in faster convergence compared with the standard SGD (i.e. without momentum). The success of momentum makes it a necessary tool for designing new optimization algorithms in optimization and deep learning. For example, all the popular variants of adaptive stochastic gradient methods like Adam ([151]) or AMSGrad ([229]) include the use of momentum.

Despite the wide use of stochastic momentum (Algorithm 19) in practice, [1] justification for the clear empirical improvements has remained elusive, as has any mathematical guidelines for actually setting the momentum parameter—it has been observed that large values (e.g. $\beta = 0.9$) work well in practice. It should be noted that Algorithm 19 is the default momentum-method in popular software packages such as PyTorch and Tensorflow. In this work we provide a theoretical analysis for SGD with momentum. We identify some mild conditions that guarantees SGD with stochastic momentum will provably escape sad-

[1] Heavy ball momentum is the default choice of momentum method in PyTorch and Tensorflow, instead of Nesterov's momentum. See the manual pages https://pytorch.org/docs/stable/_modules/torch/optim/sgd.html and https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD.

Algorithm 19 SGD with stochastic heavy ball momentum

Required: Step size parameter η and momentum parameter β.
Init: $w_0 \in \mathbb{R}^d$ and $m_{-1} = 0 \in \mathbb{R}^d$.
for $t = 0$ to T **do**
    Given current iterate w_t, obtain stochastic gradient $g_t := \nabla f(w_t; \xi_t)$.
    Update stochastic momentum $m_t := \beta m_{t-1} + g_t$.
    Update iterate $w_{t+1} := w_t - \eta m_t$.
end for

dle points faster than the standard SGD, which provides clear evidence for the benefit of using stochastic momentum. For stochastic heavy ball momentum, a weighted average of stochastic gradients at the visited points is maintained. The new update is computed as the current update minus a step in the direction of the momentum. Our analysis shows that these updates can amplify a component in an escape direction of the saddle points.

In this work, we focus on finding a second-order stationary point for smooth non-convex optimization by SGD with stochastic heavy ball momentum. Specifically, we consider the stochastic nonconvex optimization problem, $\min_{w \in \mathbb{R}^d} f(w) := \mathbb{E}_{\xi \sim \mathcal{D}}[f(w; \xi)]$, where we overload the notation so that $f(w; \xi)$ represents a stochastic function induced by the randomness ξ while $f(w)$ is the expectation of the stochastic functions. An (ϵ, ϵ)-second-order stationary point w satisfies

$$\|\nabla f(w)\| \leq \epsilon \text{ and } \nabla^2 f(w) \succeq -\epsilon I. \tag{4.1}$$

Obtaining a second order guarantee has emerged as a desired goal in the nonconvex optimization community. Since finding a global minimum or even a local minimum in general nonconvex optimization can be NP hard ([13, 213, 200, 209]), most of the papers in nonconvex optimization target at reaching an approximate second-order stationary point with additional assumptions like Lipschitzness in the gradients and the Hessian (e.g. [9, 44, 61, 63, 75, 85, 86, 105, 145, 146, 152, 167, 165, 171, 197, 212, 230, 246, 257, 282]). We follow these related works for the goal and aim at showing the benefit of the use of the momentum in reaching an (ϵ, ϵ)-second-order stationary point.

We introduce a required condition, akin to a model assumption made in (Daneshmand et al. [63]), that ensures the dynamic procedure in Algorithm 20 produces updates with suitable correlation with the negative curvature directions of the function f.

Definition 2. *Assume, at some time t, that the Hessian $H_t = \nabla^2 f(w_t)$ has some eigenvalue smaller than $-\epsilon$ and $\|\nabla f(w_t)\| \leq \epsilon$. Let v_t be the eigenvector corresponding to the smallest eigenvalue of $\nabla^2 f(w_t)$. The stochastic momentum m_t satisfies **Correlated Negative Curvature (CNC)** at t with parameter $\gamma > 0$ if*

$$\mathbb{E}_t[\langle m_t, v_t \rangle^2] \geq \gamma. \tag{4.2}$$

As we will show, the recursive dynamics of SGD with heavy ball momentum helps in amplifying the escape signal γ, which allows it to escape saddle points faster.

Contribution: We show that, under CNC assumption and some minor constraints that upper-bound parameter β, if SGD with momentum has properties called *Almost Positively Aligned with Gradient* (APAG), *Almost Positively Correlated with Gradient* (APCG), and *Gradient Alignment or Curvature Exploitation* (GrACE), defined in the later section, then it takes $T = O((1-\beta)\log(1/(1-\beta)\epsilon)\epsilon^{-10})$ iterations to return an (ϵ, ϵ) second order stationary point. Alternatively, one can obtain an $(\epsilon, \sqrt{\epsilon})$ second order stationary point in $T = O((1-\beta)\log(1/(1-\beta)\epsilon)\epsilon^{-5})$ iterations. Our theoretical result demonstrates that a larger momentum parameter β can help in escaping saddle points faster. As saddle points are pervasive in the loss landscape of optimization and deep learning ([68, 55]), the result sheds light on explaining why SGD with momentum enables training faster in optimization and deep learning.

Notation: In this chapter we use $\mathbb{E}_t[\cdot]$ to represent conditional expectation $\mathbb{E}[\cdot|w_1, w_2, \ldots, w_t]$, which is about fixing the randomness upto but not including t and notice that w_t was determined at $t-1$.

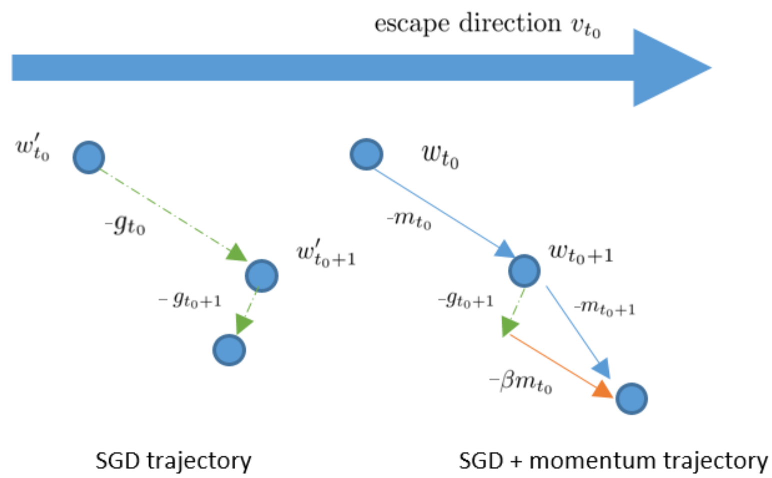


Figure 4.1: The trajectory of the standard SGD (left) and SGD with momentum (right).

4.2 Background

4.2.1 A thought experiment.

Let us provide some high-level intuition about the benefit of stochastic momentum with respect to escaping saddle points. In an iterative update scheme, at some time t_0 the parameters w_{t_0} can enter a *saddle point region*, that is a place where Hessian $\nabla^2 f(w_{t_0})$ has a non-trivial negative eigenvalue, say $\lambda_{\min}(\nabla^2 f(w_{t_0})) \leq -\epsilon$, and the gradient $\nabla f(w_{t_0})$ is small in norm, say $\|\nabla f(w_{t_0})\| \leq \epsilon$. The challenge here is that gradient updates may drift only very slowly away from the saddle point, and may not escape this region; see ([75, 165]) for additional details. On the other hand, if the iterates were to move in one particular direction, namely along v_{t_0} the direction of the smallest eigenvector of $\nabla^2 f(w_{t_0})$, then a fast escape is guaranteed under certain constraints on the step size η; see e.g. ([46]). While the negative eigenvector could be computed directly, this 2nd-order method is prohibitively expensive and hence we typically aim to rely on gradient methods. With this in mind, Daneshmand et al. [63], who study non-momentum SGD, make an assumption akin to our CNC property described above that each stochastic gradient g_{t_0} is strongly non-orthogonal to v_{t_0} the direction of large negative curvature. This suffices to drive the updates out of the saddle point region.

In the present work we study stochastic momentum, and our CNC property requires that the update direction m_{t_0} is strongly non-orthogonal to v_{t_0}; more precisely, $\mathbb{E}_{t_0}[\langle m_{t_0}, v_{t_0}\rangle^2] \geq \gamma > 0$. We are able to take advantage of the analysis of (Daneshmand et al. [63]) to establish that updates begin to escape a saddle point region for similar reasons. Further, this effect is *amplified* in successive iterations through the momentum update when β is close to 1. Assume that at some w_{t_0} we have m_{t_0} which possesses significant correlation with the negative curvature direction v_{t_0}, then on successive rounds m_{t_0+1} is quite close to βm_{t_0}, m_{t_0+2} is quite close to $\beta^2 m_{t_0}$, and so forth; see Figure 4.1 for an example. This provides an intuitive perspective on how momentum might help accelerate the escape process. Yet one might ask *does this procedure provably contribute to the escape process* and, if so, *what is the aggregate performance improvement of the momentum?* We answer the first question in the affirmative, and we answer the second question essentially by showing that momentum can help speed up saddle-point escape by a multiplicative factor of $1 - \beta$. On the negative side, we also show that β is constrained and may not be chosen arbitrarily close to 1.

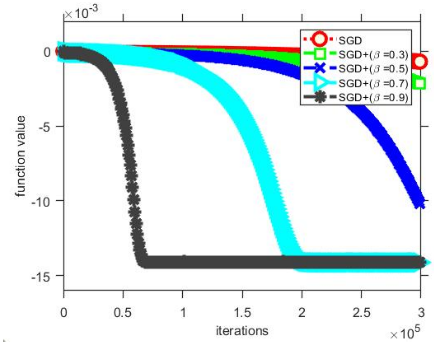
4.2.2 Momentum helps escape saddle points: an empirical view

Let us now establish, empirically, the clear benefit of stochastic momentum on the problem of saddle-point escape. We construct two stochastic optimization tasks, and each exhibits at least one significant saddle point. The two objectives are as follows.

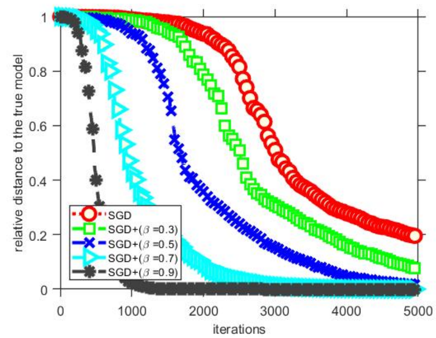
$$\min_w f(w) := \frac{1}{n}\sum_{i=1}^{n}\big(\frac{1}{2}w^\top H w + b_i^\top w + \|w\|_{10}^{10}\big), \tag{4.3}$$

$$\min_w f(w) := \frac{1}{n}\sum_{i=1}^{n}\big((a_i^\top w)^2 - y\big)^2. \tag{4.4}$$

Problem (4.3) of these was considered by (Staib et al. [246], Reddi et al. [230]) and represents a very straightforward non-convex optimization challenge, with an embedded saddle given by the matrix $H := \mathrm{diag}([1, -0.1])$, and stochastic gaussian perturbations given by



(a) Solving objective (4.3)



(b) Solving objective (4.4). (phase retrieval)

Figure 4.2: Performance of SGD with different values of $\beta = \{0, 0.3, 0.5, 0.7, 0.9\}$; $\beta = 0$ corresponds to standard SGD. Figure of the left: We plot convergence in function value $f(\cdot)$ given in (4.3). Initialization is always set as $w_0 = \mathbf{0}$. All the algorithms use the same step size $\eta = 5 \times 10^{-5}$. Figure of the right: We plot convergence in *relative distance* to the true model w^*, defined as $\min(\|w_t - w^*\|, \|w_t + w^*\|)/\|w^*\|$, which more appropriately captures progress as the global sign of the objective (4.4) is unrecoverable. All the algorithms are initialized at the same point $w_0 \sim \mathcal{N}(0, \mathcal{I}_d/(10000d))$ and use the same step size $\eta = 5 \times 10^{-4}$.

$b_i \sim \mathcal{N}(0, \text{diag}([0.1, 0.001]))$; the small variance in the second component provides lower noise in the escape direction. Here we have set $n = 10$. Observe that the origin is in the neighborhood of saddle points and has objective value zero. SGD and SGD with momentum are initialized at the origin in the experiment so that they have to escape saddle points before the convergence. The second objective (4.4) appears in the *phase retrieval* problem, that has real applications in physical sciences ([41, 240]). In phase retrieval[2], one wants to find an unknown $w^* \in \mathbb{R}^d$ with access to but a few samples $y_i = (a_i^\top w^*)^2$; the design vector a_i is known a priori. Here we have sampled $w^* \sim \mathcal{N}(0, \mathcal{I}_d/d)$ and $a_i \sim \mathcal{N}(0, \mathcal{I}_d)$ with $d = 10$ and $n = 200$.

The empirical findings, displayed in Figure 4.2, are quite stark: for both objectives, convergence is significantly accelerated by larger choices of β. In the first objective (Subfigure (a) of Figure 4.2), we see each optimization trajectory entering a saddle point region,

[2]It is known that phase retrieval is nonconvex and has the so-called strict saddle property: (1) every local minimizer $\{w^*, -w^*\}$ is global up to phase, (2) each saddle exhibits negative curvature (see e.g. ([249, 250, 52]))

apparent from the "flat" progress, yet we observe that large-momentum trajectories escape the saddle much more quickly than those with smaller momentum. A similar affect appears in Subfigure (b) of Figure 4.2. To the best of our knowledge, this is the first reported empirical finding that establishes the dramatic speed up of stochastic momentum for finding an optimal solution in phase retrieval.

4.2.3 Related works.

Heavy ball method: The heavy ball method was originally proposed by Polyak [223]. It has been observed that this algorithm, even in the deterministic setting, provides no convergence speedup over standard gradient descent, except in some highly structure cases such as convex quadratic objectives where an "accelerated" rate is possible ([168, 116, 108, 251, 183, 184, 96, 284, 150, 40]). In recent years, some works make some efforts in analyzing heavy ball method for other classes of optimization problems besides the quadratic functions. For example, Ghadimi, Feyzmahdavian, and Johansson [108] prove an $O(1/T)$ ergodic convergence rate when the problem is smooth convex, while Sun et al. [251] provide a non-ergodic convergence rate for certain classes of convex problems. Ochs et al. [214] combine the technique of forward-backward splitting with heavy ball method for a specific class of nonconvex optimization problem. For stochastic heavy ball method, Loizou and Richtárik [183] analyze a class of linear regression problems and shows a linear convergence rate of stochastic momentum, in which the linear regression problems actually belongs to the case of strongly convex quadratic functions. Other works includes ([96]), which shows almost sure convergence to the critical points by stochastic heavy ball for general non-convex coercive functions. Yet, the result does not show any advantage of stochastic heavy ball over other optimization algorithms like SGD. Can, Gürbüzbalaban, and Zhu [40] show an accelerated linear convergence to a stationary distribution under Wasserstein distance for strongly convex quadratic functions by SGD with stochastic heavy ball momentum. Yang, Lin, and Li [284] provide a unified analysis of stochastic heavy ball momentum and Nesterov's momentum for smooth non-convex objective functions. They show that the expected gradient norm converges at rate $O(1/\sqrt{t})$. Yet, the rate is not better than that of the standard SGD. We are also aware of the works of [110], [109], which propose some variants of stochastic accelerated algorithms with first order stationary point guarantees. Yet, the framework in [110, 109] does not capture the stochastic heavy ball momentum used in practice. There is also a negative result about the heavy ball momentum. Kidambi et al. [150] show that for a specific strongly convex and strongly smooth problem, SGD with heavy ball momentum fails to achieving the best convergence rate while some algorithms can.

Reaching a second order stationary point: As we mentioned earlier, there are many works aim at reaching a second order stationary point. We classify them into two categories: specialized algorithms and simple GD/SGD variants. Specialized algorithms are those designed to exploit the negative curvature explicitly and escape saddle points faster than the ones without the explicit exploitation (e.g. [46, 6, 9, 282]). Simple GD/SGD variants are those with minimal tweaks of standard GD/SGD or their variants (e.g. [105, 171, 86, 145, 147, 146, 63, 246]). Our work belongs to this category. In this category, perhaps the pioneer works are Ge et al. [105] and Jin et al. [145]. Jin et al. [145] show that explicitly adding isotropic noise in each iteration guarantees that GD escapes saddle points and finds a second order stationary point with high probability. Following Jin et al. [145], Daneshmand et al. [63] assume that stochastic gradient inherently has a component to escape. Specifically, they make assumption of the Correlated Negative Curvature (CNC) for stochastic gradient g_t so that $\mathbb{E}_t[\langle g_t, v_t \rangle^2] \geq \gamma > 0$. The assumption allows the algorithm to avoid the procedure of perturbing the updates by adding isotropic noise. Our work is motivated by Daneshmand et al. [63] but assumes CNC for the stochastic momentum m_t instead. Very recently, Jin et al. [146] consider perturbing the update of SGD and provide a second order guarantee. Staib et al. [246] consider a variant of RMSProp [256], in which the gradient g_t is multiplied by a preconditioning matrix G_t and the update is $w_{t+1} = w_t - G_t^{-1/2} g_t$.

Algorithm	Complexity
Perturbed SGD ([105])	$\mathcal{O}(\epsilon^{-16})$
Average-SGD ([86])	$\mathcal{O}(\epsilon^{-7})$
Perturbed SGD ([146])	$\mathcal{O}(\epsilon^{-8})$
CNC-SGD ([63])	$\mathcal{O}(\epsilon^{-10})$
Adaptive SGD ([246])	$\mathcal{O}(\epsilon^{-10})$
SGD+momentum (this work)	$\mathcal{O}((1-\beta)\log(\frac{1}{(1-\beta)\epsilon})\epsilon^{-10})$

Table 4.1: Iteration complexity to find an (ϵ, ϵ) second-order stationary point .

The work shows that the algorithm can help in escaping saddle points faster compared to standard SGD under certain conditions. Fang, Lin, and Zhang [86] propose average-SGD, in which a suffix averaging scheme is conducted for the updates. They also assume an inherent property of stochastic gradients that allows SGD to escape saddle points.

We summarize the iteration complexity results of the related works for simple SGD variants on Table 4.1. [3] The readers can see that the iteration complexity of Fang, Lin, and Zhang [86] and Jin et al. [146] are better than Daneshmand et al., Staib et al. [63, 246] and our result. So, we want to explain the results and clarify the differences. First, we focus on explaining why the popular algorithm, SGD with heavy ball momentum, works well in practice, which is without the suffix averaging scheme used in Fang, Lin, and Zhang [86] and is without the explicit perturbation used in Jin et al. [146]. Specifically, we focus on studying the effect of stochastic heavy ball momentum and showing the advantage of using it. Furthermore, our analysis framework is built on the work of Daneshmand et al. [63]. We believe that, based on the insight in our work, one can also show the advantage of stochastic momentum by modifying the assumptions and algorithms in ([86]) or ([146]) and consequently get a better dependency on ϵ.

[3] We follow the work Daneshmand et al. [63] for reaching an (ϵ, ϵ)-stationary point, while some works are for an $(\epsilon, \sqrt{\epsilon})$-stationary point. We translate them into the complexity of getting an (ϵ, ϵ)-stationary point.

4.3 Main Results

We assume that the gradient ∇f is L-Lipschitz; that is, f is L-smooth. Further, we assume that the Hessian $\nabla^2 f$ is ρ-Lipschitz. These two properties ensure that $\|\nabla f(w) - \nabla f(w')\| \leq L\|w - w'\|$ and that $\|\nabla^2 f(w) - \nabla^2 f(w')\| \leq \rho\|w - w'\|$, $\forall w, w'$. The L-Lipschitz gradient assumption implies that $|f(w') - f(w) - \langle \nabla f(w), w' - w\rangle| \leq \frac{L}{2}\|w - w'\|^2, \forall w, w'$, while the ρ-Lipschitz Hessian assumption implies that $|f(w') - f(w) - \langle \nabla f(w), w' - w\rangle - (w' - w)^\top \nabla^2 f(w)(w' - w)| \leq \frac{\rho}{6}\|w - w'\|^3$, $\forall w, w'$. Furthermore, we assume that the stochastic gradient has bounded noise $\|\nabla f(w) - \nabla f(w;\xi)\|^2 \leq \sigma^2$ and that the norm of stochastic momentum is bounded so that $\|m_t\| \leq c_m$. We denote $\Pi_i M_i$ as the matrix product of matrices $\{M_i\}$ and we use $\sigma_{max}(M) = \|M\|_2 := \sup_{x \neq 0} \frac{\langle x, Mx\rangle}{\langle x, x\rangle}$ to denote the spectral norm of the matrix M.

4.3.1 Required properties with empirical validation

Our analysis of stochastic momentum relies on three properties of the stochastic momentum dynamic. These properties are somewhat unusual, but we argue they should hold in natural settings, and later we aim to demonstrate that they hold empirically in a couple of standard problems of interest.

Definition 3. *We say that SGD with stochastic momentum satisfies **A**lmost **P**ositively **A**ligned with **G**radient (APAG)* [4] *if we have*

$$\mathbb{E}_t[\langle \nabla f(w_t), m_t - g_t\rangle] \geq -\frac{1}{2}\|\nabla f(w_t)\|^2. \tag{4.5}$$

*We say that SGD with stochastic momentum satisfies **A**lmost **P**ositively **C**orrelated with*

[4] Note that our analysis still go through if one replaces $\frac{1}{2}$ on r.h.s. of (4.5) with any larger number $c < 1$; the resulted iteration complexity would be only a constant multiple worse.

***Gr**adient (APCG) with parameter τ if $\exists c' > 0$ such that,*

$$\mathbb{E}_t[\langle \nabla f(w_t), M_t m_t \rangle] \geq -c' \eta \sigma_{max}(M_t) \|\nabla f(w_t)\|^2, \tag{4.6}$$

where the PSD matrix M_t is defined as

$$M_t = (\Pi_{s=1}^{\tau-1} G_{s,t})(\Pi_{s=k}^{\tau-1} G_{s,t}) \quad \textit{with} \quad G_{s,t} := I - \eta \textstyle\sum_{j=1}^{s} \beta^{s-j} \nabla^2 f(w_t) = I - \frac{\eta(1-\beta^s)}{1-\beta} \nabla^2 f(w_t)$$

for any integer $1 \leq k \leq \tau - 1$, and η is any step size chosen that guarantees each $G_{s,t}$ is PSD.

Definition 4. *We say that the SGD with momentum exhibits **Gr**adient **A**lignment or **C**urvature **E**xploitation (GrACE) if $\exists c_h \geq 0$ such that*

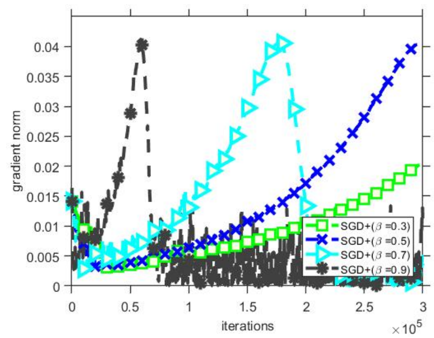
$$\mathbb{E}_t[\eta \langle \nabla f(w_t), g_t - m_t \rangle + \frac{\eta^2}{2} m_t^\top \nabla^2 f(w_t) m_t] \leq \eta^2 c_h. \tag{4.7}$$

APAG requires that the momentum term m_t must, in expectation, not be significantly misaligned with the gradient $\nabla f(w_t)$. This is a very natural condition when one sees that the momentum term is acting as a biased estimate of the gradient of the deterministic f. APAG demands that the bias can not be too large relative to the size of $\nabla f(w_t)$. Indeed this property is only needed in our analysis when the gradient is large (i.e. $\|\nabla f(w_t)\| \geq \epsilon$) as it guarantees that the algorithm makes progress; our analysis does not require APAG holds when gradient is small.

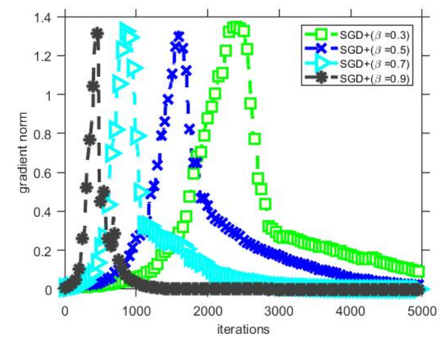
APCG is a related property, but requires that the current momentum term m_t is almost positively correlated with the the gradient $\nabla f(w_t)$, but *measured in the Mahalanobis norm induced by M_t*. It may appear to be an unusual object, but one can view the PSD matrix M_t as measuring something about the local curvature of the function with respect to the trajectory of the SGD with momentum dynamic. We will show that this property holds empirically on two natural problems for a reasonable constant c'. APCG is only needed in our analysis when the update is in a saddle region with significant negative curvature, $\|\nabla f(w)\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w)) \leq -\epsilon$. Our analysis does not require APCG holds when the gradient is large or the update is at an (ϵ, ϵ)-second order stationary point.

For GrACE, the first term on l.h.s of (4.7) measures the alignment between stochastic momentum m_t and the gradient $\nabla f(w_t)$, while the second term on l.h.s measures the curvature exploitation. The first term is small (or even negative) when the stochastic momentum m_t is aligned with the gradient $\nabla f(w_t)$, while the second term is small (or even negative) when the stochastic momentum m_t can exploit a negative curvature (i.e. the subspace of eigenvectors that corresponds to the negative eigenvalues of the Hessian $\nabla^2 f(w_t)$ if exists). Overall, a small sum of the two terms (and, consequently, a small c_h) allows one to bound the function value of the next iterate (see Lemma 43).

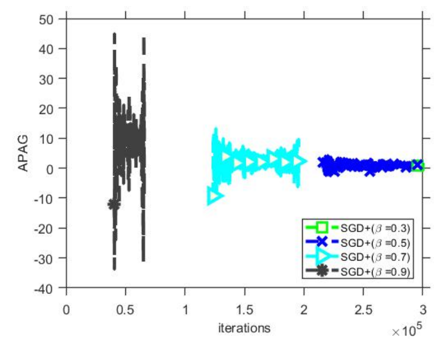
On Figure 4.3, we report some quantities related to APAG and APCG as well as the gradient norm when solving the previously discussed problems (4.3) and (4.4) using SGD with momentum. We also report a quantity regarding GrACE on Figure 4.4.



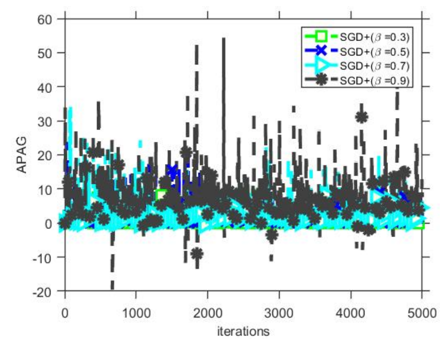
(a) Gradient norm $\|\nabla f(w_t)\|$.



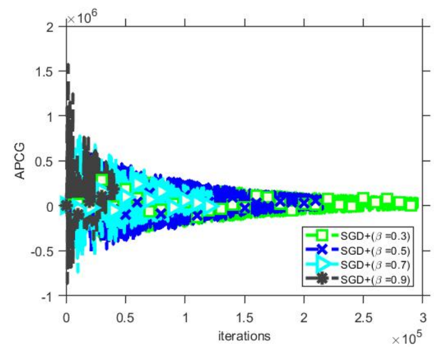
(b) Gradient norm $\|\nabla f(w_t)\|$.



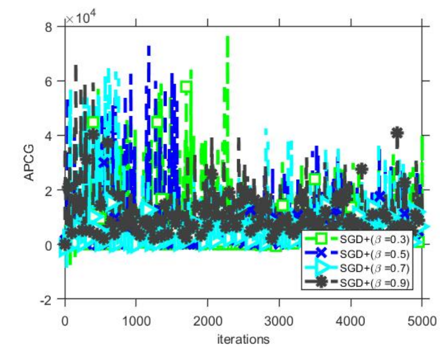
(c) About APAG.



(d) About APAG.

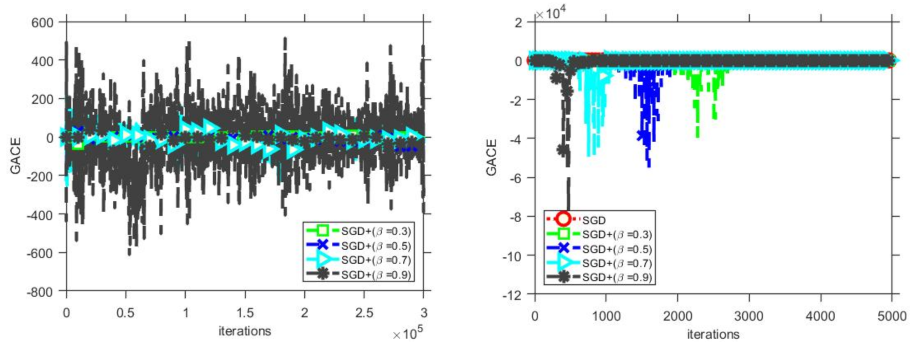


(e) About APCG.



(f) About APCG.

Figure 4.3: Plots of the related properties. Sub-figures on the left are regarding solving (4.3) and sub-figures on the right are regarding solving (4.4) (phase retrieval). Note that the function value/relative distance to w^* are plotted on Figure 4.2. Above, sub-figures (a) and (b): We plot the gradient norms versus iterations. Sub-figures (c) and (d): We plot the values of $\langle \nabla f(w_t), m_t - g_t \rangle / \|\nabla f(w_t)\|^2$ versus iterations. For (c), we only report them when the gradient is large ($\|\nabla f(w_t)\| \geq 0.02$). It shows that the value is large than -0.5 except the transition. For (d), we observe that the value is almost always nonnegative. Sub-figures (e) and (f): We plot the value of $\langle \nabla f(w_t), M_t m_t \rangle / (\eta \sigma_{max}(M_t) \|\nabla f(w_t)\|^2)$. For (e), we let $M_t = (\Pi_{s=1}^{3\times 10^5} G_{s,t})(\Pi_{s=1}^{3\times 10^5} G_{s,t})$ and we only report the values when the update is in the region of saddle points. For (f), we let $M_t = (\Pi_{s=1}^{500} G_{s,t})(\Pi_{s=1}^{500} G_{s,t})$ and we observe that the value is almost always nonnegative. The figures implies that SGD with momentum has APAG and APCG properties in the experiments. Furthermore, an interesting observation is that, for the phase retrieval problem, the expected values might actually be nonnegative.



(a) About GrACE for problem (4.3). (b) About GrACE for problem (4.4). (phase retrieval)

Figure 4.4: Plot regarding the GrACE property. We plot the values of $\big(\eta\langle\nabla f(w_t), g_t - m_t\rangle + \frac{1}{2}\eta^2 m_t^\top H_t m_t\big)/\eta^2$ versus iterations. An interesting observation is that the value is well upper-bounded by zero for the phase retrieval problem. The results imply that the constant c_h is indeed small.

4.3.2 Convergence results

The high level idea of our analysis follows as a similar template to ([145, 63, 246]). Our proof is structured into three cases: either (a) $\|\nabla f(w)\| \geq \epsilon$, or (b) $\|\nabla f(w)\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w)) \leq -\epsilon$, or otherwise (c) $\|\nabla f(w)\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w)) \geq -\epsilon$, meaning we have arrived in a second-order stationary region. The precise algorithm we analyze is Algorithm 20, which identical to Algorithm 19 except that we boost the step size to a larger value r on occasion. We will show that the algorithm makes progress in cases (a) and (b). In case (c), when the goal has already been met, further execution of the algorithm only weakly hurts progress. Ultimately, we prove that a second order stationary point is arrived at with high probability. While our proof borrows tools from ([63, 246]), much of the momentum analysis is entirely novel to our knowledge.

Theorem 29. *Assume that the stochastic momentum satisfies CNC. Set* [5] $r = O(\epsilon^2)$, $\eta = O(\epsilon^5)$, *and* $\mathcal{T}_{thred} = \frac{c(1-\beta)}{\eta\epsilon}\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon}) = O((1-\beta)\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon})\epsilon^{-6})$ *for some constant* $c > 0$. *If SGD with momentum (Algorithm 20) has APAG property when gradi-*

[5] See Table 4.2 in Section 4.4.4 for the precise expressions of the parameters. Here, we hide the parameters' dependencies on γ, L, c_m, c', σ^2, ρ, c_h, and δ. W.l.o.g, we also assume that c_m, L, σ^2, c', c_h, and ρ are not less than one and $\epsilon \leq 1$.

Algorithm 20 SGD with stochastic heavy ball momentum

1: Required: Step size parameters r and η, momentum parameter β, and period parameter $\mathcal{T}_{thred}$.

2: Init: $w_0 \in \mathbb{R}^d$ and $m_{-1} = 0 \in \mathbb{R}^d$.
3: **for** $t = 0$ to T **do**
4: Get stochastic gradient g_t at w_t, and set stochastic momentum $m_t := \beta m_{t-1} + g_t$.
5: Set learning rate: $\hat{\eta} := \eta$ **unless** $(t \bmod \mathcal{T}_{thred}) = 0$ in which case $\hat{\eta} := r$
6: $w_{t+1} = w_t - \hat{\eta} m_t$.
7: **end for**

ent is large ($\|\nabla f(w)\| \geq \epsilon$), APCG$_{\mathcal{T}_{thred}}$ property when it enters a region of saddle points that exhibits a negative curvature ($\|\nabla f(w)\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w)) \leq -\epsilon$), and GrACE property throughout the iterations, then it reaches an (ϵ, ϵ) second order stationary point in $T = 2\mathcal{T}_{thred}(f(w_0) - \min_w f(w))/(\delta \mathcal{F}_{thred}) = O((1-\beta)\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon})\epsilon^{-10})$ iterations with high probability $1 - \delta$, where $\mathcal{F}_{thred} = O(\epsilon^4)$.

The theorem implies the advantage of using stochastic momentum for SGD. Higher β leads to reaching a second order stationary point faster. As we will show in the following, this is due to that higher β enables escaping the saddle points faster. In Subsection 3.2.1, we provide some key details of the proof of Theorem 29. The interested reader can read a high-level sketch of the proof, as well as the detailed version, in Section 4.4.6.

Remark 1: (constraints on β) .

W.l.o.g, we assume that c_m, L, σ^2, c', c_h, and ρ are not less than one and that $\epsilon \leq 1$. [6] We require that parameter β is not too close to 1 so that the following holds,

- 1) $L(1-\beta)^3 > 1$.
- 2) $\sigma^2(1-\beta)^3 > 1$.
- 3) $c'(1-\beta)^2 > 1$.
- 4) $\eta \leq \frac{1-\beta}{L}$.
- 5) $\eta \leq \frac{1-\beta}{\epsilon}$.

[6] We assume that β is chosen so that $1-\beta$ is not too small and consequently the choice of η satisfies $\eta \leq \min\{\frac{(1-\beta)}{L}, \frac{(1-\beta)}{\epsilon}\}$.

Table 4.2: Constraints and choices of the parameters.

Parameter	Value	Constraint origin	constant
r	$\delta\gamma\epsilon^2 c_r$	(4.64), (4.65), (4.66)	$c_r \leq \frac{c_0}{c_m^3 \rho L \sigma^2 c_h}$, $c_0 = \frac{1}{1152}$ $\frac{c_0}{c_m^3 \rho L \sigma^2 c' (1-\beta)^2 c_h} \leq c_r$ $\frac{c_0}{c_m^3 \rho L \sigma^4 (1-\beta)^3 c_h} \leq c_r$
r	”	$r \leq \sqrt{\frac{\delta \mathcal{F}_{thred}}{8 c_h}}$ from (4.90)	”
η	$\delta^2\gamma^2\epsilon^5 c_\eta$	(4.64)	$c_\eta \leq \frac{c_1}{c_m^5 \rho L^2 \sigma^2 c' c_h}$, $c_1 = \frac{c_0}{24}$
η	”	$\eta \leq r/\sqrt{\mathcal{T}_{thred}}$ from (4.25),(4.39),(4.88),(4.90)	”
η	”	$\eta \leq \min\{\frac{(1-\beta)}{L}, \frac{(1-\beta)}{\epsilon}\}$ from (4.45), (4.78)	
$\mathcal{F}_{thred}$	$\delta\gamma^2\epsilon^4 c_F$	(4.65)	$c_F \leq \frac{c_2}{c_m^4 \rho^2 L \sigma^4 c_h}$, $c_2 = \frac{c_0}{576}$ $c_F \geq \frac{8 c_0^2}{c_m^6 \rho^2 L^2 \sigma^4 c_h}$
$\mathcal{F}_{thred}$	”	$\mathcal{F}_{thred} \leq \frac{\epsilon^2 r}{4}$ from (4.89)	”
$\mathcal{T}_{thred}$		$\mathcal{T}_{thred} \geq \frac{c(1-\beta)}{\eta\epsilon} \log(\frac{L c_m \sigma^2 \rho c' c_h}{(1-\beta)\delta\gamma\epsilon})$ from (4.82)	

- 6) $\mathcal{T}_{thred} \geq \frac{c(1-\beta)}{\eta\epsilon} \log(\frac{L c_m \sigma^2 \rho c' c_h}{(1-\beta)\delta\gamma\epsilon}) \geq 1 + \frac{2\beta}{1-\beta}$.

The constraints upper-bound the value of β. That is, β cannot be too close to 1. We note that the β dependence on L, σ, and c' are only artificial. We use these constraints in our proofs but they are mostly artefacts of the analysis. For example, if a function is L-smooth, and $L < 1$, then it is also 1-smooth, so we can assume without loss of generality that $L > 1$. Similarly, the dependence on σ is not highly relevant, since we can always increase the variance of the stochastic gradient, for example by adding an $O(1)$ gaussian perturbation.

Remark 2: (escaping saddle points) Note that Algorithm 2 reduces to CNC-SGD of [63] when $\beta = 0$ (i.e. without momentum). Therefore, let us compare the results. We show that the escape time of Algorithm 2 is $T_{thred} := \tilde{O}\big(\frac{(1-\beta)}{\eta\epsilon}\big)$ (see Section 4.4.4, especially (4.81-4.82)). On the other hand, for CNC-SGD, based on Table 3 in their paper, is $T_{thred} = \tilde{O}\big(\frac{1}{\eta\epsilon}\big)$. One can clearly see that T_{thred} of our result has a dependency $1-\beta$, which makes it smaller than that of [63] for any same η and consequently demonstrates escaping saddle point faster with momentum.

Remark 3: (finding a second order stationary point) Denote ℓ a number such that

$\forall t, \|g_t\| \leq \ell$. In the following, we will show that in the high momentum regime where $(1-\beta) \ll \frac{\rho^2 \ell^{10}}{c_m^9 c_h^2 c'}$, Algorithm 20 is strictly better than CNC-SGD of [63], which means that a higher momentum can help find a second order stationary point faster. Empirically, we find out that $c' \approx 0$ (Figure 4.3) and $c_h \approx 0$ (Figure 4.4) in the phase retrieval problem, so the condition is easily satisfied for a wide range of β.

Comparison to [63] Theorem 2 in [63] states that, for CNC-SGD to find an $(\epsilon, \rho^{1/2}\epsilon)$ stationary point, the total number of iterations is

$$T = O(\frac{\ell^{10} L^3}{\delta^4 \gamma^4} \log^2(\frac{\ell L}{\epsilon \delta \gamma}) \epsilon^{-10}),$$

where ℓ is the bound of the stochastic gradient norm $\|g_t\| \leq \ell$ which can be viewed as the counterpart of c_m in our work. By translating their result for finding an (ϵ, ϵ) stationary point, it is $T = O(\frac{\ell^{10} L^3 \rho^5}{\delta^4 \gamma^4} \log^2(\frac{\rho \ell L}{\epsilon \delta \gamma}) \epsilon^{-10})$. On the other hand, using the parameters value on Table 4.2, we have that $T = 2\mathcal{T}_{thred}\big(f(w_0) - \min_w f(w)\big)/(\delta \mathcal{F}_{thred}) = O(\frac{(1-\beta) c_m^9 L^3 \rho^3 (\sigma^2)^3 c_h^2 c'}{\delta^4 \gamma^4} \log(\frac{L c_m \sigma^2 c' c_h}{(1-\beta)\delta \gamma \epsilon}) \epsilon^{-10})$ for Algorithm 20.

Before making a comparison, we note that their result does not have a dependency on the variance of stochastic gradient (i.e. σ^2), which is because they assume that the variance is also bounded by the constant ℓ (can be seen from (86) in the supplementary of their paper where the variance terms $\|\zeta_i\|$ are bounded by ℓ). Following their treatment, if we assume that $\sigma^2 \leq c_m$, then on (4.71) we can instead replace $(\sigma^2 + 3c_m^2)$ with $4c_m^2$ and on (4.72) it becomes $1 \geq \frac{576 c_m^3 \rho c_r \epsilon}{(1-\beta)^3}$. This will remove all the parameters' dependency on σ^2. Now by comparing $\tilde{O}((1-\beta) c_m^9 c_h^2 c' \cdot \frac{\rho^3 L^3}{\delta^4 \gamma^4} \epsilon^{-10})$ of ours and $T = \tilde{O}(\rho^2 \ell^{10} \cdot \frac{\rho^3 L^3}{\delta^4 \gamma^4} \epsilon^{-10})$ of [63], we see that in the high momentum regime where $(1-\beta) << \frac{\rho^2 \ell^{10}}{c_m^9 c_h^2 c'}$, Algorithm 20 is strictly better than that of [63], which means that a higher momentum can help to find a second order stationary point faster.

4.3.3 Escaping saddle points

In this subsection, we analyze the process of escaping saddle points by SGD with momentum. Denote t_0 any time such that $(t_0 \mod \mathcal{T}_{thred}) = 0$. Suppose that it enters the region exhibiting a small gradient but a large negative eigenvalue of the Hessian (i.e. $\|\nabla f(w_{t_0})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w_{t_0})) \leq -\epsilon$). We want to show that it takes at most $\mathcal{T}_{thred}$ iterations to escape the region and whenever it escapes, the function value decreases at least by $\mathcal{F}_{thred} = O(\epsilon^4)$ on expectation, where the precise expression of $\mathcal{F}_{thred}$ will be determined later in Section 4.4.4. The technique that we use is proving by contradiction. Assume that the function value on expectation does not decrease at least $\mathcal{F}_{thred}$ in $\mathcal{T}_{thred}$ iterations. Then, we get an upper bound of the expected distance $\mathbb{E}_{t_0}[\|w_{t_0+\mathcal{T}_{thred}} - w_{t_0}\|^2] \leq C_{\text{upper}}$. Yet, by leveraging the negative curvature, we also show a lower bound of the form $\mathbb{E}_{t_0}[\|w_{t_0+\mathcal{T}_{thred}} - w_{t_0}\|^2] \geq C_{\text{lower}}$. The analysis will show that the lower bound is larger than the upper bound (namely, $C_{\text{lower}} > C_{\text{upper}}$), which leads to the contradiction and concludes that the function value must decrease at least $\mathcal{F}_{thred}$ in $\mathcal{T}_{thred}$ iterations on expectation. Since $\mathcal{T}_{thred} = O((1-\beta)\log(\frac{1}{(1-\beta)\epsilon})\epsilon^6)$, the dependency on β suggests that larger β can leads to smaller $\mathcal{T}_{thred}$, which implies that larger momentum helps in escaping saddle points faster.

Lemma 36 below provides an upper bound of the expected distance. The proof is in Section 4.4.2.

Lemma 36. *Denote t_0 any time such that $(t_0 \mod \mathcal{T}_{thred}) = 0$. Suppose that $\mathbb{E}_{t_0}[f(w_{t_0}) - f(w_{t_0+t})] \leq \mathcal{F}_{thred}$ for any $0 \leq t \leq \mathcal{T}_{thred}$. Then, $\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \leq C_{upper,t} := \frac{8\eta t\left(\mathcal{F}_{thred}+2r^2c_h+\frac{\rho}{3}r^3c_m^3\right)}{(1-\beta)^2} + 8\eta^2\frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2\left(\frac{\beta}{1-\beta}\right)^2 c_m^2 + 2r^2c_m^2$.*

We see that $C_{\text{upper,t}}$ in Lemma 36 is monotone increasing with t, so we can define $C_{\text{upper}} := C_{\text{upper},\mathcal{T}_{thred}}$. Now let us switch to obtaining the lower bound of $\mathbb{E}_{t_0}[\|w_{t_0+\mathcal{T}_{thred}} - w_{t_0}\|^2]$. The key to get the lower bound comes from the recursive dynamics of SGD with momentum.

Lemma 37. *Denote t_0 any time such that $(t_0 \mod \mathcal{T}_{thred}) = 0$. Let us define a quadratic approximation at w_{t_0}, $Q(w) := f(w_{t_0}) + \langle w - w_{t_0}, \nabla f(w_{t_0})\rangle + \frac{1}{2}(w - w_{t_0})^\top H(w - w_{t_0})$, where $H := \nabla^2 f(w_{t_0})$. Also, define $G_s := (I - \eta \sum_{k=1}^{s} \beta^{s-k} H)$. Then we can write $w_{t_0+t} - w_{t_0}$ exactly using the following decomposition.*

$$\begin{aligned}
&\overbrace{\big(\Pi_{j=1}^{t-1} G_j\big)\big(-rm_{t_0}\big)}^{q_{v,t-1}} + \overbrace{\eta\,(-1)\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1} G_j\big)\beta^s m_{t_0}}^{q_{m,t-1}} \\
+\;& \overbrace{\eta\,(-1)\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1} G_j\big)\sum_{k=1}^{s}\beta^{s-k}\big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big)}^{q_{q,t-1}} \\
+\;& \overbrace{\eta\,(-1)\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1} G_j\big)\sum_{k=1}^{s}\beta^{s-k}\nabla f(w_{t_0})}^{q_{w,t-1}} + \overbrace{\eta\,(-1)\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1} G_j\big)\sum_{k=1}^{s}\beta^{s-k}\xi_{t_0+k}}^{q_{\xi,t-1}}.
\end{aligned}$$

The proof of Lemma 37 is in Section 4.4.3. Furthermore, we will use the quantities $q_{v,t-1}, q_{m,t-1}, q_{q,t-1}, q_{w,t-1}, q_{\xi,t-1}$ as defined above throughout the analysis.

Lemma 38. *Following the notations of Lemma 37, we have that*

$$\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \geq \mathbb{E}_{t_0}[\|q_{v,t-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{m,t-1} + q_{q,t-1} + q_{w,t-1} + q_{\xi,t-1}\rangle] =: C_{lower}.$$

We are going to show that the dominant term in the lower bound of $\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2]$ is $\mathbb{E}_{t_0}[\|q_{v,t-1}\|^2]$, which is the critical component for ensuring that the lower bound is larger than the upper bound of the expected distance.

Lemma 39. *Denote $\theta_j := \sum_{k=1}^{j} \beta^{j-k} = \sum_{k=1}^{j} \beta^{k-1}$ and $\lambda := -\lambda_{\min}(H)$. Following the conditions and notations in Lemma 36 and Lemma 37, we have that*

$$\mathbb{E}_{t_0}[\|q_{v,t-1}\|^2] \geq \big(\Pi_{j=1}^{t-1}(1 + \eta\theta_j\lambda)\big)^2 r^2\gamma. \tag{4.8}$$

Proof. We know that $\lambda_{\min}(H) \leq -\epsilon < 0$. Let v be the eigenvector of the Hessian H with unit norm that corresponds to $\lambda_{\min}(H)$ so that $Hv = \lambda_{\min}(H)v$. We have $(I - \eta H)v = v - \eta\lambda_{\min}(H)v = (1 - \eta\lambda_{\min}(H))v$. Then,

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|q_{v,t-1}\|^2] \overset{(a)}{=} \mathbb{E}_{t_0}[\|q_{v,t-1}\|^2\|v\|^2] \overset{(b)}{\geq} \mathbb{E}_{t_0}[\langle q_{v,t-1}, v\rangle^2] \overset{(c)}{=} \mathbb{E}_{t_0}[\langle \big(\Pi_{j=1}^{t-1} G_j\big) r m_{t_0}, v\rangle^2] \\
&\overset{(d)}{=} \mathbb{E}_{t_0}[\langle \big(\Pi_{j=1}^{t-1}(I - \eta\theta_j H)\big) r m_{t_0}, v\rangle^2] = \mathbb{E}_{t_0}\langle \big(\Pi_{j=1}^{t-1}(1 - \eta\theta_j\lambda_{\min}(H))\big) r m_{t_0}, v\rangle^2] \\
&\overset{(e)}{\geq} \big(\Pi_{j=1}^{t-1}(1 + \eta\theta_j\lambda)\big)^2 r^2\gamma,
\end{aligned} \tag{4.9}$$

where (a) is because v is with unit norm, (b) is by Cauchy–Schwarz inequality, (c), (d) are by the definitions, and (e) is by the CNC assumption so that $\mathbb{E}_{t_0}[\langle m_{t_0}, v\rangle^2] \geq \gamma$. □

Observe that the lower bound in (4.8) is monotone increasing with t and the momentum parameter β. Moreover, it actually grows exponentially in t. To get the contradiction, we have to show that the lower bound is larger than the upper bound. By Lemma 36 and Lemma 38, it suffices to prove the following lemma. We provide its proof in Section 4.4.4.

Lemma 40. *Let $\mathcal{F}_{thred} = O(\epsilon^4)$ and $\eta^2\mathcal{T}_{thred} \leq r^2$. By following the conditions and notations in Theorem 29, Lemma 36 and Lemma 37, we conclude that if SGD with momentum (Algorithm 20) has the APCG property, then we have that $C_{lower} := \mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{m,\mathcal{T}_{thred}-1} + q_{q,\mathcal{T}_{thred}-1} + q_{w,\mathcal{T}_{thred}-1} + q_{\xi,\mathcal{T}_{thred}-1}\rangle] > C_{upper}$.*

4.4 Detailed proofs

4.4.1 Lemma 41, 42, and 43

In the following, Lemma 42 says that under the APAG property, when the gradient norm is large, on expectation SGD with momentum decreases the function value by a constant and consequently makes progress. On the other hand, Lemma 43 upper-bounds the increase of function value of the next iterate (if happens) by leveraging the GrACE property.

Lemma 41. *If SGD with momentum has the APAG property, then, considering the update step $w_{t+1} = w_t - \eta m_t$, we have that $\mathbb{E}_t[f(w_{t+1})] \leq f(w_t) - \frac{\eta}{2}\|\nabla f(w_t)\|^2 + \frac{L\eta^2 c_m^2}{2}$.*

Proof. By the L-smoothness assumption,

$$\begin{aligned} f(w_{t+1}) &\leq f(w_t) - \eta\langle \nabla f(w_t), m_t\rangle + \frac{L\eta^2}{2}\|m_t\|^2 \\ &\leq f(w_t) - \eta\langle \nabla f(w_t), g_t\rangle - \eta\langle \nabla f(w_t), m_t - g_t\rangle + \frac{L\eta^2 c_m^2}{2}. \end{aligned} \tag{4.10}$$

Taking the expectation on both sides. We have

$$\begin{aligned} &\mathbb{E}_t[f(w_{t+1})] \leq f(w_t) - \eta\|\nabla f(w_t)\|^2 - \eta\mathbb{E}_t[\langle \nabla f(w_t), m_t - g_t\rangle] + \frac{L\eta^2 c_m^2}{2} \\ &\leq f(w_t) - \frac{\eta}{2}\|\nabla f(w_t)\|^2 + \frac{L\eta^2 c_m^2}{2}. \end{aligned} \tag{4.11}$$

where we use the APAG property in the last inequality.

□

Lemma 42. *Assume that the step size η satisfies $\eta \leq \frac{\epsilon^2}{8Lc_m^2}$. If SGD with momentum has the APAG property, then, considering the update step $w_{t+1} = w_t - \eta m_t$, we have that $\mathbb{E}_t[f(w_{t+1})] \leq f(w_t) - \frac{\eta}{4}\epsilon^2$ when $\|\nabla f(w_t)\| \geq \epsilon$.*

Proof. $\mathbb{E}_t[f(w_{t+1}) - f(w_t)] \overset{Lemma\ 41}{\leq} -\frac{\eta}{2}\|\nabla f(w_t)\|^2 + \frac{L\eta^2 c_m^2}{2} \overset{\|\nabla f(w_t)\| \geq \epsilon}{\leq} -\frac{\eta}{2}\epsilon^2 + \frac{L\eta^2 c_m^2}{2} \leq -\frac{\eta}{4}\epsilon^2$, where the last inequality is due to the constraint of η. □

Lemma 43. *If SGD with momentum has the GrACE property, then, considering the update step $w_{t+1} = w_t - \eta m_t$, we have that $\mathbb{E}_t[f(w_{t+1})] \leq f(w_t) + \eta^2 c_h + \frac{\rho\eta^3}{6} c_m^3$.*

Proof. Consider the update rule $w_{t+1} = w_t - \eta m_t$, where m_t represents the stochastic momentum and η is the step size. By ρ-Lipschitzness of Hessian, we have $f(w_{t+1}) \leq f(w_t) - \eta\langle \nabla f(w_t), g_t\rangle + \eta\langle \nabla f(w_t), g_t - m_t\rangle + \frac{\eta^2}{2} m_t^\top \nabla^2 f(w_t) m_t + \frac{\rho\eta^3}{6}\|m_t\|^3$. Taking the conditional expectation, one has

$$\begin{aligned}
\mathbb{E}_t[f(w_{t+1})] \leq f(w_t) - \mathbb{E}_t[\eta\|\nabla f(w_t)\|^2] + \mathbb{E}_t[\eta\langle\nabla f(w_t), g_t - m_t\rangle + \frac{\eta^2}{2} m_t^\top \nabla^2 f(w_t) m_t] \\
+ \frac{\rho\eta^3}{6} c_m^3. \\
\leq f(w_t) + 0 + \eta^2 c_h + \frac{\rho\eta^3}{6} c_m^3.
\end{aligned} \tag{4.12}$$

□

4.4.2 Proof of Lemma 36

Lemma 36 *Denote* t_0 *any time such that* $(t_0 \mod \mathcal{T}_{thred}) = 0$. *Suppose that* $\mathbb{E}_{t_0}[f(w_{t_0}) - f(w_{t_0+t})] \leq \mathcal{F}_{thred}$ *for any* $0 \leq t \leq \mathcal{T}_{thred}$. *Then,*

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \\
&\leq C_{upper,t} \\
&:= \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\eta^2 \frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2.
\end{aligned} \tag{4.13}$$

Proof. Recall that the update is $w_{t_0+1} = w_{t_0} - r m_{t_0}$, and $w_{t_0+t} = w_{t_0+t-1} - \eta m_{t_0+t-1}$, for $t > 1$. We have that

$$\|w_{t_0+t} - w_{t_0}\|^2 \leq 2(\|w_{t_0+t} - w_{t_0+1}\|^2 + \|w_{t_0+1} - w_{t_0}\|^2) \leq 2\|w_{t_0+t} - w_{t_0+1}\|^2 + 2r^2 c_m^2, \tag{4.14}$$

where the first inequality is by the triangle inequality and the second one is due to the assumption that $\|m_t\| \leq c_m$ for any t. Now let us denote

- $\alpha_s := \sum_{j=0}^{t-1-s} \beta^j$
- $A_{t-1} := \sum_{s=1}^{t-1} \alpha_s$

and let us rewrite $g_t = \nabla f(w_t) + \xi_t$, where ξ_t is the zero-mean noise. We have that

$$
\begin{aligned}
&\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0+1}\|^2] = \mathbb{E}_{t_0}[\|\sum_{s=1}^{t-1} -\eta m_{t_0+s}\|^2] = \mathbb{E}_{t_0}[\eta^2\|\sum_{s=1}^{t-1}\big((\sum_{j=1}^{s}\beta^{s-j}g_{t_0+j}) + \beta^s m_{t_0}\big)\|^2] \\
&\leq \mathbb{E}_{t_0}[2\eta^2\|\sum_{s=1}^{t-1}\sum_{j=1}^{s}\beta^{s-j}g_{t_0+j}\|^2 + 2\eta^2\|\sum_{s=1}^{t-1}\beta^s m_{t_0}\|^2] \\
&\leq \mathbb{E}_{t_0}[2\eta^2\|\sum_{s=1}^{t-1}\sum_{j=1}^{s}\beta^{s-j}g_{t_0+j}\|^2] + 2\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 \\
&= \mathbb{E}_{t_0}[2\eta^2\|\sum_{s=1}^{t-1}\alpha_s g_{t_0+s}\|^2] + 2\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 \\
&= \mathbb{E}_{t_0}[2\eta^2\|\sum_{s=1}^{t-1}\alpha_s\big(\nabla f(w_{t_0+s}) + \xi_{t_0+s}\big)\|^2] + 2\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 \\
&\leq \mathbb{E}_{t_0}[4\eta^2\|\sum_{s=1}^{t-1}\alpha_s\nabla f(w_{t_0+s})\|^2] + \mathbb{E}_{t_0}[4\eta^2\|\sum_{s=1}^{t-1}\alpha_s\xi_{t_0+s}\|^2] + 2\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2.
\end{aligned}
\tag{4.15}
$$

To proceed, we need to upper bound $\mathbb{E}_{t_0}[4\eta^2\|\sum_{s=1}^{t-1}\alpha_s\nabla f(w_{t_0+s})\|^2]$. We have that

$$
\begin{aligned}
\mathbb{E}_{t_0}[4\eta^2\|\sum_{s=1}^{t-1}\alpha_s\nabla f(w_{t_0+s})\|^2] &\overset{(a)}{\leq} \mathbb{E}_{t_0}[4\eta^2 A_{t-1}^2\sum_{s=1}^{t-1}\frac{\alpha_s}{A_{t-1}}\|\nabla f(w_{t_0+s})\|^2] \\
&\overset{(b)}{\leq} \mathbb{E}_{t_0}[4\eta^2\frac{A_{t-1}}{1-\beta}\sum_{s=1}^{t-1}\|\nabla f(w_{t_0+s})\|^2] \\
&\overset{(c)}{\leq} \mathbb{E}_{t_0}[4\eta^2\frac{t}{(1-\beta)^2}\sum_{s=1}^{t-1}\|\nabla f(w_{t_0+s})\|^2].
\end{aligned}
\tag{4.16}
$$

where (a) is by Jensen's inequality, (b) is by $\max_s \alpha_s \leq \frac{1}{1-\beta}$, and (c) is by $A_{t-1} \leq \frac{t}{1-\beta}$.

Now let us switch to bound the other term.

$$\begin{aligned}
&\mathbb{E}_{t_0}[4\eta^2\|\sum_{s=1}^{t-1}\alpha_s\xi_{t_0+s}\|^2] = 4\eta^2\big(\mathbb{E}_{t_0}[\sum_{i\neq j}\alpha_i\alpha_j\xi_{t_0+i}^\top\xi_{t_0+j}] + \mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\alpha_s^2\xi_{t_0+s}^\top\xi_{t_0+s}]\big)\\
&\overset{(a)}{=} 4\eta^2\big(0 + \mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\alpha_s^2\xi_{t_0+s}^\top\xi_{t_0+s}]\big),\\
&\overset{(b)}{\leq} 4\eta^2\frac{t\sigma^2}{(1-\beta)^2}.
\end{aligned} \tag{4.17}$$

where (a) is because $\mathbb{E}_{t_0}[\xi_{t_0+i}^\top\xi_{t_0+j}] = 0$ for $i \neq j$, (b) is by that $\|\xi_t\|^2 \leq \sigma^2$ and $\max_t \alpha_t \leq \frac{1}{1-\beta}$. Combining (4.14), (4.15), (4.16), (4.17),

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2]\\
&\leq \mathbb{E}_{t_0}[8\eta^2\frac{t}{(1-\beta)^2}\sum_{s=1}^{t-1}\|\nabla f(w_{t_0+s})\|^2] + 8\eta^2\frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2c_m^2.
\end{aligned} \tag{4.18}$$

Now we need to bound $\mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\|\nabla f(w_{t_0+s})\|^2]$. By using ρ-Lipschitzness of Hessian, we have that

$$\begin{aligned}
f(w_{t_0+s}) \leq\ & f(w_{t_0+s-1}) - \eta\langle\nabla f(w_{t_0+s-1}), m_{t_0+s-1}\rangle + \frac{1}{2}\eta^2 m_{t_0+s-1}^\top\nabla^2 f(w_{t_0+s-1})m_{t_0+s-1}\\
&+ \frac{\rho}{6}\eta^3\|m_{t_0+s-1}\|^3.
\end{aligned} \tag{4.19}$$

By adding $\eta\langle\nabla f(w_{t_0+s-1}), g_{t_0+s-1}\rangle$ on both sides, we have

$$\begin{aligned}
\eta\langle\nabla f(w_{t_0+s-1}), g_{t_0+s-1}\rangle \leq\ & f(w_{t_0+s-1}) - f(w_{t_0+s}) + \eta\langle\nabla f(w_{t_0+s-1}), g_{t_0+s-1} - m_{t_0+s-1}\rangle\\
&+ \frac{1}{2}\eta^2 m_{t_0+s-1}^\top\nabla^2 f(w_{t_0+s-1})m_{t_0+s-1} + \frac{\rho}{6}\eta^3\|m_{t_0+s-1}\|^3.
\end{aligned} \tag{4.20}$$

Taking conditional expectation on both sides leads to

$$\mathbb{E}_{t_0+s-1}[\eta\|\nabla f(w_{t_0+s-1})\|^2] \leq \mathbb{E}_{t_0+s-1}[f(w_{t_0+s-1}) - f(w_{t_0+s})] + \eta^2 c_h + \frac{\rho}{6}\eta^3 c_m^3, \quad (4.21)$$

where $\mathbb{E}_{t_0+s-1}[\eta\langle\nabla f(w_{t_0+s-1}), g_{t_0+s-1} - m_{t_0+s-1}\rangle + \frac{1}{2}\eta^2 m_{t_0+s-1}^\top \nabla^2 f(w_{t_0+s-1}) m_{t_0+s-1}] \leq \eta^2 c_h$ by the GrACE property. We have that for $t_0 \leq t_0 + s - 1$

$$\begin{aligned}
&\mathbb{E}_{t_0}[\eta\|\nabla f(w_{t_0+s-1})\|^2] = \mathbb{E}_{t_0}[\mathbb{E}_{t_0+s-1}[\eta\|\nabla f(w_{t_0+s-1})\|^2]] \\
&\overset{(4.21)}{\leq} \mathbb{E}_{t_0}[\mathbb{E}_{t_0+s-1}[f(w_{t_0+s-1}) - f(w_{t_0+s})]] + \eta^2 c_h + \frac{\rho}{6}\eta^3 c_m^3 \\
&= \mathbb{E}_{t_0}[f(w_{t_0+s-1}) - f(w_{t_0+s})] + \eta^2 c_h + \frac{\rho}{6}\eta^3 c_m^3.
\end{aligned} \quad (4.22)$$

Summing the above inequality from $s = 2, 3, \ldots, t$ leads to

$$\begin{aligned}
&\mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\eta\|\nabla f(w_{t_0+s})\|^2] \leq \mathbb{E}_{t_0}[f(w_{t_0+1}) - f(w_{t_0+t})] + \eta^2(t-1)c_h + \frac{\rho}{6}\eta^3(t-1)c_m^3 \\
&= \mathbb{E}_{t_0}[f(w_{t_0+1}) - f(w_{t_0}) + f(w_{t_0}) - f(w_{t_0+t})] + \eta^2(t-1)c_h + \frac{\rho}{6}\eta^3(t-1)c_m^3 \\
&\overset{(a)}{\leq} \mathbb{E}_{t_0}[f(w_{t_0+1}) - f(w_{t_0})] + \mathcal{F}_{thred} + \eta^2(t-1)c_h + \frac{\rho}{6}\eta^3(t-1)c_m^3,
\end{aligned} \quad (4.23)$$

where (a) is by the assumption (made for proving by contradiction) that $\mathbb{E}_{t_0}[f(w_{t_0}) - f(w_{t_0+s})] \leq \mathcal{F}_{thred}$ for any $0 \leq s \leq \mathcal{T}_{thred}$. By (4.21) with $s = 1$ and $\eta = r$, we have

$$\mathbb{E}_{t_0}[r\|\nabla f(w_{t_0})\|^2] \leq \mathbb{E}_{t_0}[f(w_{t_0}) - f(w_{t_0+1})] + r^2 c_h + \frac{\rho}{6}r^3 c_m^3. \quad (4.24)$$

By (4.23) and (4.24), we know that

$$\begin{aligned}
&\mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\eta\|\nabla f(w_{t_0+s})\|^2] \leq \mathbb{E}_{t_0}[r\|f(w_{t_0})\|^2] + \mathbb{E}_{t_0}[\sum_{s=1}^{t-1}\eta\|\nabla f(w_{t_0+s})\|^2] \\
&\leq \mathcal{F}_{thred} + r^2 c_h + \frac{\rho}{6}r^3 c_m^3 + \eta^2 t c_h + \frac{\rho}{6}\eta^3 t c_m^3 \\
&\overset{(a)}{\leq} \mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{6}r^3 c_m^3 + \frac{\rho}{6}r^2\eta c_m^3. \\
&\overset{(b)}{\leq} \mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3}r^3 c_m^3,
\end{aligned} \tag{4.25}$$

where (a) is by the constraint that $\eta^2 t \leq r^2$ for $0 \leq t \leq \mathcal{T}_{thred}$ and (b) is by the constraint that $r \geq \eta$. By combining (4.25) and (4.18)

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \\
&\leq \mathbb{E}_{t_0}[8\eta^2 \frac{t}{(1-\beta)^2}\sum_{s=1}^{t-1}\|\nabla f(w_{t_0+s})\|^2] + 8\eta^2\frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2 \\
&\leq \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3}r^3 c_m^3\big)}{(1-\beta)^2} + 8\eta^2\frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2.
\end{aligned} \tag{4.26}$$

□

4.4.3 Proof of Lemma 37 and Lemma 38

Lemma 37 *Denote t_0 any time such that $(t_0 \mod \mathcal{T}_{thred}) = 0$. Let us define a quadratic approximation at w_{t_0}, $Q(w) := f(w_{t_0}) + \langle w - w_{t_0}, \nabla f(w_{t_0})\rangle + \frac{1}{2}(w - w_{t_0})^\top H(w - w_{t_0})$, where $H := \nabla^2 f(w_{t_0})$. Also, define $G_s := (I - \eta\sum_{k=1}^{s}\beta^{s-k}H)$ and*

- $q_{v,t-1} := \big(\Pi_{j=1}^{t-1}G_j\big)\big(-rm_{t_0}\big)$.
- $q_{m,t-1} := -\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1}G_j\big)\beta^s m_{t_0}$.
- $q_{q,t-1} := -\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1}G_j\big)\sum_{k=1}^{s}\beta^{s-k}\big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big)$.
- $q_{w,t-1} := -\sum_{s=1}^{t-1}\big(\Pi_{j=s+1}^{t-1}G_j\big)\sum_{k=1}^{s}\beta^{s-k}\nabla f(w_{t_0})$.

- $q_{\xi,t-1} := -\sum_{s=1}^{t-1} \left(\Pi_{j=s+1}^{t-1} G_j\right) \sum_{k=1}^{s} \beta^{s-k} \xi_{t_0+k}$.

Then, $w_{t_0+t} - w_{t_0} = q_{v,t-1} + \eta q_{m,t-1} + \eta q_{q,t-1} + \eta q_{w,t-1} + \eta q_{\xi,t-1}$.

Notations:

Denote t_0 any time such that $(t_0 \mod \mathcal{T}_{thred}) = 0$. Let us define a quadratic approximation at w_{t_0},

$$Q(w) := f(w_{t_0}) + \langle w - w_{t_0}, \nabla f(w_{t_0}) \rangle + \frac{1}{2}(w - w_{t_0})^\top H (w - w_{t_0}), \tag{4.27}$$

where $H := \nabla^2 f(w_{t_0})$. Also, we denote

$$\begin{aligned}
G_s &:= (I - \eta \sum_{k=1}^{s} \beta^{s-k} H) \\
v_{m,s} &:= \beta^s m_{t_0} \\
v_{q,s} &:= \sum_{k=1}^{s} \beta^{s-k} \big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big) \\
v_{w,s} &:= \sum_{k=1}^{s} \beta^{s-k} \nabla f(w_{t_0}) \\
v_{\xi,s} &:= \sum_{k=1}^{s} \beta^{s-k} \xi_{t_0+k} \\
\theta_s &:= \sum_{k=1}^{s} \beta^{s-k}.
\end{aligned} \tag{4.28}$$

Proof. First, we rewrite m_{t_0+j} for any $j \geq 1$ as follows.

$$\begin{aligned}
m_{t_0+j} &= \beta^j m_{t_0} + \sum_{k=1}^{j} \beta^{j-k} g_{t_0+k} \\
&= \beta^j m_{t_0} + \sum_{k=1}^{j} \beta^{j-k} \big(\nabla f(w_{t_0+k}) + \xi_{t_0+k}\big).
\end{aligned} \tag{4.29}$$

We have that

$$
\begin{aligned}
w_{t_0+t} - w_{t_0} &= w_{t_0+t-1} - w_{t_0} - \eta m_{t_0+t-1} \\
&\overset{(a)}{=} w_{t_0+t-1} - w_{t_0} - \eta\big(\beta^{t-1} m_{t_0} + \sum_{k=1}^{t-1} \beta^{t-1-k}\big(\nabla f(w_{t_0+k}) + \xi_{t_0+k}\big)\big) \\
&\overset{(b)}{=} w_{t_0+t-1} - w_{t_0} - \eta \sum_{k=1}^{t-1} \beta^{t-1-k} \nabla Q(w_{t_0+t-1}) \\
&- \eta\big(\beta^{t-1} m_{t_0} + \sum_{k=1}^{t-1} \beta^{t-1-k}\big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+t-1}) + \xi_{t_0+k}\big)\big) \\
&\overset{(c)}{=} w_{t_0+t-1} - w_{t_0} - \eta \sum_{k=1}^{t-1} \beta^{t-1-k}\big(H(w_{t_0+t-1} - w_{t_0}) + \nabla f(w_{t_0})\big) \\
&- \eta\big(\beta^{t-1} m_{t_0} + \sum_{k=1}^{t-1} \beta^{t-1-k}\big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+t-1}) + \xi_{t_0+k}\big)\big) \\
&= (I - \eta \sum_{k=1}^{t-1} \beta^{t-1-k} H)\big(w_{t_0+t-1} - w_{t_0}\big) \\
&- \eta\big(\beta^{t-1} m_{t_0} + \sum_{k=1}^{t-1} \beta^{t-1-k}\big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+t-1}) + \nabla f(w_{t_0}) + \xi_{t_0+k}\big)\big),
\end{aligned}
\tag{4.30}
$$

where (a) is by using (4.29) with $j = t-1$, (b) is by subtracting and adding back the same term, and (c) is by $\nabla Q(w_{t_0+t-1}) = \nabla f(w_{t_0}) + H(w_{t_0+t-1} - w_{t_0})$.

To continue, by using the nations in (4.28), we can rewrite (4.30) as

$$
w_{t_0+t} - w_{t_0} = G_{t-1}\big(w_{t_0+t-1} - w_{t_0}\big) - \eta\big(v_{m,t-1} + v_{q,t-1} + v_{w,t-1} + v_{\xi,t-1}\big). \tag{4.31}
$$

Recursively expanding (4.31) leads to

$$\begin{aligned}
w_{t_0+t} - w_{t_0} &= G_{t-1}\big(w_{t_0+t-1} - w_{t_0}\big) - \eta\big(v_{m,t-1} + v_{q,t-1} + v_{w,t-1} + v_{\xi,t-1}\big) \\
&= G_{t-1}\big(G_{t-2}\big(w_{t_0+t-2} - w_{t_0}\big) - \eta\big(v_{m,t-2} + v_{q,t-2} + v_{w,t-2} + v_{\xi,t-2}\big)\big) \\
&- \eta\big(v_{m,t-1} + v_{q,t-1} + v_{w,t-1} + v_{\xi,t-1}\big) \\
&\overset{(a)}{=} \big(\Pi_{j=1}^{t-1} G_j\big)\big(w_{t_0+1} - w_{t_0}\big) - \eta \sum_{s=1}^{t-1} \big(\Pi_{j=s+1}^{t-1} G_j\big)\big(v_{m,s} + v_{q,s} + v_{w,s} + v_{\xi,s}\big), \\
&\overset{(b)}{=} \big(\Pi_{j=1}^{t-1} G_j\big)\big(-rm_{t_0}\big) - \eta \sum_{s=1}^{t-1} \big(\Pi_{j=s+1}^{t-1} G_j\big)\big(v_{m,s} + v_{q,s} + v_{w,s} + v_{\xi,s}\big),
\end{aligned} \tag{4.32}$$

where (a) we use the notation that $\Pi_{j=s}^{t-1} G_j := G_s \times G_{s+1} \times \ldots\ldots G_{t-1}$ and the notation that $\Pi_{j=t}^{t-1} G_j = 1$ and (b) is by the update rule. By using the definitions of $\{q_{\star,t-1}\}$ in the lemma statement, we complete the proof.

□

Lemma 38 *Following the notations of Lemma 37, we have that*

$$\begin{aligned}
\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] &\geq \mathbb{E}_{t_0}[\|q_{v,t-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{m,t-1} + q_{q,t-1} + q_{w,t-1} + q_{\xi,t-1}\rangle] \\
&:= C_{\mathit{lower}}
\end{aligned} \tag{4.33}$$

Proof. Following the proof of Lemma 37, we have

$$w_{t_0+t} - w_{t_0} = q_{v,t-1} + \eta\big(q_{m,t-1} + q_{q,t-1} + q_{w,t-1} + q_{\xi,t-1}\big). \tag{4.34}$$

Therefore, by using $\|a+b\|^2 \geq \|a\|^2 + 2\langle a, b\rangle$,

$$\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \geq \mathbb{E}_{t_0}[\|q_{v,t-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{m,t-1} + q_{q,t-1} + q_{w,t-1} + q_{\xi,t-1}\rangle]. \tag{4.35}$$

□

4.4.4 Proof of Lemma 40

Lemma 40 *Let $\mathcal{F}_{thred} = O(\epsilon^4)$ and $\eta^2 \mathcal{T}_{thred} \leq r^2$. By following the conditions and notations in Theorem 29, Lemma 36 and Lemma 37, we conclude that if SGD with momentum (Algorithm 20) has the APCG property, then we have that $C_{lower} := \mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta \mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{m,\mathcal{T}_{thred}-1} + q_{q,\mathcal{T}_{thred}-1} + q_{w,\mathcal{T}_{thred}-1} + q_{\xi,\mathcal{T}_{thred}-1}\rangle] > C_{upper}$.*

Some supporting lemmas To prove Lemma 40, we need a series of lemmas with the choices of parameters on Table 4.2.

Upper bounding $\mathbb{E}_{t_0}[\|q_{q,t-1}\|]$:

Lemma 44. *Following the conditions in Lemma 36 and Lemma 37, we have*

$$\begin{aligned}\mathbb{E}_{t_0}[\|q_{q,t-1}\|] \leq & \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)\frac{\beta L c_m}{\epsilon(1-\beta)^2} \\ & + \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta}\frac{\rho}{\eta\epsilon^2}\frac{8\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} \\ & + \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta}\frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon}.\end{aligned} \tag{4.36}$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{t_0}[\|q_{q,t-1}\|] = \mathbb{E}_{t_0}[\| - \sum_{s=1}^{t-1} \big(\Pi_{j=s+1}^{t-1} G_j\big) \sum_{k=1}^{s} \beta^{s-k} \big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big) \|] \\
&\overset{(a)}{\leq} \mathbb{E}_{t_0}[\sum_{s=1}^{t-1} \| \big(\Pi_{j=s+1}^{t-1} G_j\big) \sum_{k=1}^{s} \beta^{s-k} \big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big) \|] \\
&\overset{(b)}{\leq} \mathbb{E}_{t_0}[\sum_{s=1}^{t-1} \| \big(\Pi_{j=s+1}^{t-1} G_j\big) \|_2 \| \sum_{k=1}^{s} \beta^{s-k} \big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big) \|] \\
&\overset{(c)}{\leq} \mathbb{E}_{t_0}[\sum_{s=1}^{t-1} \| \big(\Pi_{j=s+1}^{t-1} G_j\big) \|_2 \sum_{k=1}^{s} \beta^{s-k} \| \big(\nabla f(w_{t_0+k}) - \nabla Q(w_{t_0+s})\big) \|] \\
&\overset{(d)}{\leq} \mathbb{E}_{t_0}[\sum_{s=1}^{t-1} \| \big(\Pi_{j=s+1}^{t-1} G_j\big) \|_2 \sum_{k=1}^{s} \beta^{s-k} \big(\|\nabla f(w_{t_0+k}) - \nabla f(w_{t_0+s})\| \\
&\qquad + \|\nabla f(w_{t_0+s}) - \nabla Q(w_{t_0+s})\| \big)],
\end{aligned}
\tag{4.37}
$$

where (a), (c), (d) is by triangle inequality, (b) is by the fact that $\|Ax\|_2 \leq \|A\|_2 \|x\|_2$ for any matrix A and vector x. Now that we have an upper bound of $\|\nabla f(w_{t_0+k}) - \nabla f(w_{t_0+s})\|$,

$$
\|\nabla f(w_{t_0+k}) - \nabla f(w_{t_0+s})\| \overset{(a)}{\leq} L\|w_{t_0+k} - w_{t_0+s}\| \overset{(b)}{\leq} L\eta(s-k)c_m. \tag{4.38}
$$

where (a) is by the assumption of L-Lipschitz gradient and (b) is by applying the triangle inequality $(s-k)$ times and that $\|w_t - w_{t-1}\| \leq \eta \|m_{t-1}\| \leq \eta c_m$, for any t. We can also derive an upper bound of $\mathbb{E}_{t_0}[\|\nabla f(w_{t_0+s}) - \nabla Q(w_{t_0+s})\|]$,

$$
\begin{aligned}
&\mathbb{E}_{t_0}[\|\nabla f(w_{t_0+s}) - \nabla Q(w_{t_0+s})\|] \\
&\overset{(a)}{\leq} \mathbb{E}_{t_0}[\frac{\rho}{2}\|w_{t_0+s} - w_{t_0}\|^2] \\
&\overset{(b)}{\leq} \frac{\rho}{2}\Big(\frac{8\eta s\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2 . \Big)
\end{aligned}
\tag{4.39}
$$

Above, (a) is by the fact that if a function $f(\cdot)$ has ρ Lipschitz Hessian, then

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\| \leq \frac{\rho}{2}\|y-x\|^2 \tag{4.40}$$

(c.f. Lemma 1.2.4 in ([211])) and using the definition that

$$Q(w) := f(w_{t_0}) + \langle w - w_{t_0}, \nabla f(w_{t_0}) \rangle + \frac{1}{2}(w - w_{t_0})^\top H (w - w_{t_0}),$$

(b) is by Lemma 36 and $\eta^2 t \leq r^2$ for $0 \leq t \leq \mathcal{T}_{thred}$

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|w_{t_0+t} - w_{t_0}\|^2] \\
&\leq \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\eta^2 \frac{t\sigma^2}{(1-\beta)^2} + 4\eta^2 \big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2 \\
&\leq \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8 \frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2 \big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2.
\end{aligned} \tag{4.41}$$

Combing (4.37), (4.38), (4.39), we have that

$$\begin{aligned}
&\mathbb{E}_{t_0}[\|q_{q,t-1}\|] \\
&\overset{(4.37)}{\leq} \mathbb{E}_{t_0}\Big[\sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \big(\|\nabla f(w_{t_0+k}) - \nabla f(w_{t_0+s})\| \\
&\qquad + \|\nabla f(w_{t_0+s}) - \nabla Q(w_{t_0+s})\|\big)\Big] \\
&\overset{(4.38),(4.39)}{\leq} \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} L\eta(s-k) c_m \\
&+ \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \frac{\rho}{2} \Big(\frac{8\eta s\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\frac{r^2\sigma^2}{(1-\beta)^2} \\
&+ 4\eta^2 \big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2 \Big) \\
&:= \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} L\eta(s-k) c_m + \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \frac{\rho}{2}(\nu_s + \nu),
\end{aligned} \tag{4.42}$$

where on the last line we use the notation that

$$\begin{aligned}\nu_s &:= \frac{8\eta s\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} \\ \nu &:= 8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2.\end{aligned} \tag{4.43}$$

To continue, let us analyze $\|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2$ first.

$$\begin{aligned}\|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 &= \|\Pi_{j=s+1}^{t-1}(I - \eta\sum_{k=1}^{j}\beta^{j-k}H)\|_2 \\ &\overset{(a)}{\leq} \Pi_{j=s+1}^{t-1}(1+\eta\theta_j\lambda) = \frac{\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)}{\Pi_{j=1}^{s}(1+\eta\theta_j\lambda)} \overset{(b)}{\leq} \frac{\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)}{(1+\eta\epsilon)^s}.\end{aligned} \tag{4.44}$$

Above, we use the notation that $\theta_j := \sum_{k=1}^{j}\beta^{j-k}$. For (a), it is due to that $\lambda := -\lambda_{min}(H)$, $\lambda_{\max}(H) \leq L$, and the choice of η so that $1 \geq \frac{\eta L}{1-\beta}$, or equivalently,

$$\eta \leq \frac{1-\beta}{L}. \tag{4.45}$$

For (b), it is due to that $\theta_j \geq 1$ for any j and $\lambda \geq \epsilon$. Therefore, we can upper-bound the first term on r.h.s of (4.42) as

$$\begin{aligned}&\sum_{s=1}^{t-1}\|\big(\Pi_{j=s+1}^{t-1}G_j\big)\|_2\sum_{k=1}^{s}\beta^{s-k}L\eta(s-k)c_m = \sum_{s=1}^{t-1}\|\big(\Pi_{j=s+1}^{t-1}G_j\big)\|_2\sum_{k=1}^{s-1}\beta^k kL\eta c_m \\ &\overset{(a)}{\leq} \sum_{s=1}^{t-1}\|\big(\Pi_{j=s+1}^{t-1}G_j\big)\|_2\frac{\beta}{(1-\beta)^2}L\eta c_m \\ &\overset{(b)}{\leq} \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)\frac{\beta L\eta c_m}{(1-\beta)^2}\sum_{s=1}^{t-1}\frac{1}{(1+\eta\epsilon)^s} \\ &\overset{(c)}{\leq} \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)\frac{\beta L\eta c_m}{(1-\beta)^2}\frac{1}{\eta\epsilon} = \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)\frac{\beta L c_m}{\epsilon(1-\beta)^2},\end{aligned} \tag{4.46}$$

where (a) is by that fact that $\sum_{k=1}^{\infty}\beta^k k \leq \frac{\beta}{(1-\beta)^2}$ for any $0 \leq \beta < 1$, (b) is by using (4.44), and (c) is by using that $\sum_{s=1}^{\infty}(\frac{1}{1+\eta\epsilon})^s \leq \frac{1}{\eta\epsilon}$. Now let us switch to bound

$\sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \frac{\rho}{2}(\nu_s+\nu)$ on (4.42). We have that

$$
\begin{aligned}
&\sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \frac{\rho}{2}(\nu_s+\nu) \overset{(a)}{\leq} \frac{1}{1-\beta} \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \frac{\rho}{2}(\nu_s+\nu) \\
&\overset{(b)}{\leq} \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \sum_{s=1}^{t-1} \frac{1}{(1+\eta\epsilon)^s} \frac{\rho}{2}\nu_s + \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \sum_{s=1}^{t-1} \frac{1}{(1+\eta\epsilon)^s} \frac{\rho}{2}\nu \\
&\overset{(c)}{\leq} \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \sum_{s=1}^{t-1} \frac{1}{(1+\eta\epsilon)^s} \frac{\rho}{2}\nu_s + \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho\nu}{2\eta\epsilon} \\
&= \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \sum_{s=1}^{t-1} \frac{1}{(1+\eta\epsilon)^s} \frac{\rho}{2}\nu_s \\
&\quad + \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon} \\
&\overset{(d)}{\leq} \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho}{(\eta\epsilon)^2} \frac{8\eta\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} \\
&+ \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon}
\end{aligned}
\tag{4.47}
$$

where (a) is by the fact that $\sum_{k=1}^{s} \beta^{s-k} \leq 1/(1-\beta)$, (b) is by (4.44), (c) is by using that $\sum_{s=1}^{\infty} (\frac{1}{1+\eta\epsilon})^s \leq \frac{1}{\eta\epsilon}$, (d) is by $\sum_{k=1}^{\infty} z^k k \leq \frac{z}{(1-z)^2}$ for any $|z| \leq 1$ and substituting $z = \frac{1}{1+\eta\epsilon}$, which leads to $\sum_{k=1}^{\infty} z^k k \leq \frac{z}{(1-z)^2} = \frac{1/(1+\eta\epsilon)}{(1-1/(1+\eta\epsilon))^2} = \frac{1+\eta\epsilon}{(\eta\epsilon)^2} \leq \frac{2}{(\eta\epsilon)^2}$ in which the last inequality is by chosen the step size η so that $\eta\epsilon \leq 1$.

By combining (4.42), (4.46), and (4.47), we have that

$$
\begin{aligned}
\mathbb{E}_{t_0}[\|q_{q,t-1}\| &\overset{(4.42)}{\leq} \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} L\eta(s-k) c_m \\
&\quad + \sum_{s=1}^{t-1} \|\big(\Pi_{j=s+1}^{t-1} G_j\big)\|_2 \sum_{k=1}^{s} \beta^{s-k} \frac{\rho}{2}(\nu_s+\nu) \\
&\overset{(4.46),(4.47)}{\leq} \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big) \frac{\beta L c_m}{\epsilon(1-\beta)^2} \\
&+ \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho}{\eta\epsilon^2} \frac{8\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} \\
&+ \frac{\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)}{1-\beta} \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon},
\end{aligned}
\tag{4.48}
$$

which completes the proof.

$\square$

Upper bounding $\|q_{v,t-1}\|$:

Lemma 45. *Following the conditions in Lemma 36 and Lemma 37, we have*

$$\|q_{v,t-1}\| \leq \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)rc_m. \tag{4.49}$$

Proof.

$$\|q_{v,t-1}\| \leq \|\big(\Pi_{j=1}^{t-1}G_j\big)\big(-rm_{t_0}\big)\| \leq \|\big(\Pi_{j=1}^{t-1}G_j\big)\|_2\|-rm_{t_0}\| \leq \big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)rc_m, \tag{4.50}$$

where the last inequality is because η is chosen so that $1 \geq \frac{\eta L}{1-\beta}$ and the fact that $\lambda_{\max}(H) \leq L$.

$\square$

Lower bounding $\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{q,t-1}\rangle]$:

Lemma 46. *Following the conditions in Lemma 36 and Lemma 37, we have*

$$\begin{aligned}
&\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{q,t-1}\rangle] \\
&\geq -2\eta\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)^2 rc_m\times \\
&[\frac{\beta Lc_m}{\epsilon(1-\beta)^2} + \frac{\rho}{\eta\epsilon^2}\frac{8\big(\mathcal{F}_{thred}+2r^2c_h+\frac{\rho}{3}r^3c_m^3\big)}{(1-\beta)^3} + \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2}+4\eta^2\big(\frac{\beta}{1-\beta}\big)^2c_m^2+2r^2c_m^2\big)}{2\eta\epsilon(1-\beta)}].
\end{aligned} \tag{4.51}$$

Proof. By the results of Lemma 44 and Lemma 45

$$
\begin{aligned}
&\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{q,t-1}\rangle] \geq -\mathbb{E}_{t_0}[2\eta\|q_{v,t-1}\|\|q_{q,t-1}\|] \\
&\overset{Lemma\ 45}{\geq} -\mathbb{E}_{t_0}[2\eta\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)rc_m\|q_{q,t-1}\|] \\
&\overset{Lemma\ 44}{\geq} -2\eta\big(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\big)^2 rc_m\times \\
&[\frac{\beta L c_m}{\epsilon(1-\beta)^2} + \frac{\rho}{\eta\epsilon^2}\frac{8\big(\mathcal{F}_{thred}+2r^2c_h+\frac{\rho}{3}r^3c_m^3\big)}{(1-\beta)^3} + \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2}+4\eta^2\big(\frac{\beta}{1-\beta}\big)^2c_m^2+2r^2c_m^2\big)}{2\eta\epsilon(1-\beta)}].
\end{aligned}
\tag{4.52}
$$

□

Lower bounding $\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{\xi,t-1}\rangle]$:

Lemma 47. *Following the conditions in Lemma 36 and Lemma 37, we have*

$$\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{\xi,t-1}\rangle] = 0. \tag{4.53}$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{\xi,t-1}\rangle] = \mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, -\sum_{s=1}^{t-1}\left(\Pi_{j=s+1}^{t-1}G_j\right)\sum_{k=1}^{s}\beta^{s-k}\xi_{t_0+k}\rangle] \\
&\overset{(a)}{=} \mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, \sum_{k=1}^{s}\alpha_k\xi_{t_0+k}\rangle] \\
&\overset{(b)}{=} \mathbb{E}_{t_0}[2\eta\sum_{k=1}^{s}\mathbb{E}_{t_0+k-1}[\langle q_{v,t-1}, \alpha_k\xi_{t_0+k}\rangle]] \\
&\overset{(c)}{=} \mathbb{E}_{t_0}[2\eta\sum_{k=1}^{s}\langle q_{v,t-1}, \mathbb{E}_{t_0+k-1}[\alpha_k\xi_{t_0+k}]\rangle] \\
&= \mathbb{E}_{t_0}[2\eta\sum_{k=1}^{s}\alpha_k\langle q_{v,t-1}, \mathbb{E}_{t_0+k-1}[\xi_{t_0+k}]\rangle] \\
&\overset{(d)}{=} 0,
\end{aligned}
\tag{4.54}
$$

where (a) holds for some coefficients α_k, (b) is by the tower rule, (c) is because $q_{v,t-1}$ is measureable with t_0, and (d) is by the zero mean assumption of ξ's.

□

Lower bounding $\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{m,t-1}\rangle]$:

Lemma 48. *Following the conditions in Lemma 36 and Lemma 37, we have*

$$\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{m,t-1}\rangle] \geq 0. \tag{4.55}$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{t_0}[2\eta\langle q_{v,t-1}, q_{m,t-1}\rangle] \\
&= 2\eta r \mathbb{E}_{t_0}[\langle \big(\Pi_{j=1}^{t-1} G_j\big) m_{t_0}, \sum_{s=1}^{t-1} \big(\Pi_{j=s+1}^{t-1} G_j\big)\beta^s m_{t_0}\rangle] \\
&\overset{(a)}{=} 2\eta r \mathbb{E}_{t_0}[\langle m_{t_0}, B m_{t_0}\rangle] \overset{(b)}{\geq} 0,
\end{aligned} \tag{4.56}$$

where (a) is by defining the matrix $B := \big(\Pi_{j=1}^{t-1} G_j\big)^\top \big(\sum_{s=1}^{t-1} \big(\Pi_{j=s+1}^{t-1} G_j\big)\beta^s\big)$. For (b), notice that the matrix B is symmetric positive semidefinite. To see that the matrix B is symmetric positive semidefinite, observe that each $G_j := (I - \eta \sum_{k=1}^{j} \beta^{j-k} H)$ can be written in the form of $G_j = U D_j U^\top$ for some orthonormal matrix U and a diagonal matrix D_j. Therefore, the matrix product $\big(\Pi_{j=1}^{t-1} G_j\big)^\top \big(\Pi_{j=s+1}^{t-1} G_j\big) = U(\Pi_{j=1}^{t-1} D_j)(\Pi_{j=s+1}^{t-1} D_j) U^\top$ is symmetric positive semidefinite as long as each G_j is. So, (b) is by the property of a matrix being symmetric positive semidefinite.

□

Lower bounding $2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{w,t-1}\rangle]$:

Lemma 49. *Following the conditions in Lemma 36 and Lemma 37, if SGD with momentum has the APCG property, then*

$$2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{w,t-1}\rangle] \geq -\frac{2\eta r c'}{(1-\beta)} (\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda))^2 \epsilon. \tag{4.57}$$

Proof. Define $D_s := \Pi_{j=1}^{t-1} G_j \Pi_{j=s+1}^{t-1} G_j$.

$$
\begin{aligned}
&2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{w,t-1}\rangle] = 2\eta\mathbb{E}_{t_0}[\langle \left(\Pi_{j=1}^{t-1} G_j\right)\left(rm_{t_0}\right), \sum_{s=1}^{t-1} \left(\Pi_{j=s+1}^{t-1} G_j\right) \sum_{k=1}^{s} \beta^{s-k} \nabla f(w_{t_0})\rangle] \\
&= 2\eta\mathbb{E}_{t_0}[\langle rm_{t_0}, \sum_{s=1}^{t-1} \left(\Pi_{j=1}^{t-1} G_j \Pi_{j=s+1}^{t-1} G_j\right) \sum_{k=1}^{s} \beta^{s-k} \nabla f(w_{t_0})\rangle] \\
&= 2\eta r \sum_{s=1}^{t-1} \sum_{k=1}^{s} \beta^{s-k} \mathbb{E}_{t_0}[\langle m_{t_0}, D_s \nabla f(w_{t_0})\rangle] \\
&\overset{(a)}{\geq} -2\eta^2 rc' \sum_{s=1}^{t-1} \sum_{k=1}^{s} \beta^{s-k} \|D_s\|_2 \|\nabla f(w_{t_0})\|^2 \\
&\geq -\frac{2\eta^2 rc'}{1-\beta} \sum_{s=1}^{t-1} \|D_s\|_2 \|\nabla f(w_{t_0})\|^2,
\end{aligned}
\tag{4.58}
$$

where (a) is by the APCG property. We also have that

$$
\begin{aligned}
\|D_s\|_2 &= \|\Pi_{j=1}^{t-1} G_j \Pi_{j=s+1}^{t-1} G_j\|_2 \leq \|\Pi_{j=1}^{t-1} G_j\|_2 \|\Pi_{j=s+1}^{t-1} G_j\|_2 \\
&\overset{(a)}{\leq} \|\Pi_{j=1}^{t-1} G_j\|_2 \frac{\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)}{(1+\eta\epsilon)^s} \overset{(b)}{\leq} \frac{\left(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\right)^2}{(1+\eta\epsilon)^s}
\end{aligned}
\tag{4.59}
$$

where (a) and (b) is by (4.44). Substituting the result back to (4.58), we get

$$
\begin{aligned}
2\eta\mathbb{E}_{t_0}[\langle q_{v,t-1}, q_{w,t-1}\rangle] &\geq -\frac{2\eta^2 rc'}{1-\beta} \sum_{s=1}^{t-1} \|D_s\|_2 \|\nabla f(w_{t_0})\|^2 \\
&\geq -\frac{2\eta^2 rc'}{1-\beta} \sum_{s=1}^{t-1} \frac{\left(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\right)^2}{(1+\eta\epsilon)^s} \|\nabla f(w_{t_0})\|^2 \\
&\geq -\frac{2\eta^2 rc'}{(1-\beta)\eta\epsilon} \left(\Pi_{j=1}^{t-1}(1+\eta\theta_j\lambda)\right)^2 \|\nabla f(w_{t_0})\|^2
\end{aligned}
\tag{4.60}
$$

Using the fact that $\|\nabla f(w_{t_0})\| \leq \epsilon$ completes the proof.

$\square$

Proof of Lemma 40 Recall that the strategy is proving by contradiction. Assume that the function value does not decrease at least $\mathcal{F}_{thred}$ in $\mathcal{T}_{thred}$ iterations on expectation. Then, we can get an upper bound of the expected distance $\mathbb{E}_{t_0}[\|w_{t_0+\mathcal{T}_{thred}} - w_{t_0}\|^2] \leq C_{\text{upper}}$ but, by leveraging the negative curvature, we can also show a lower bound of the form $\mathbb{E}_{t_0}[\|w_{t_0+\mathcal{T}_{thred}} - w_{t_0}\|^2] \geq C_{\text{lower}}$. The strategy is showing that the lower bound is larger than the upper bound, which leads to the contradiction and concludes that the function value must decrease at least $\mathcal{F}_{thred}$ in $\mathcal{T}_{thred}$ iterations on expectation. To get the contradiction, according to Lemma 36 and Lemma 38, we need to show that

$$\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{m,\mathcal{T}_{thred}-1} + q_{q,\mathcal{T}_{thred}-1} + q_{w,\mathcal{T}_{thred}-1} + q_{\xi,\mathcal{T}_{thred}-1}\rangle] \\ > C_{upper}. \tag{4.61}$$

Yet, by Lemma 48 and Lemma 47, we have that $\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{m,\mathcal{T}_{thred}-1}\rangle] \geq 0$ and $\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{\xi,\mathcal{T}_{thred}-1}\rangle] = 0$. So, it suffices to prove that

$$\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{q,\mathcal{T}_{thred}-1} + q_{w,\mathcal{T}_{thred}-1}\rangle] > C_{upper}, \tag{4.62}$$

and it suffices to show that

- $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{q,\mathcal{T}_{thred}-1}\rangle] \geq 0.$
- $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{w,\mathcal{T}_{thred}-1}\rangle] \geq 0.$
- $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] \geq C_{upper}.$

Proving that $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta\mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{q,\mathcal{T}_{thred}-1}\rangle] \geq 0$:

By Lemma 39 and Lemma 46, we have that

$$\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + \mathbb{E}_{t_0}[2\eta\langle q_{v,\mathcal{T}_{thred}-1}, q_{q,\mathcal{T}_{thred}-1}\rangle]$$
$$\geq \frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda)\big)^2 r^2\gamma - 2\eta\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda)\big)^2 r c_m$$
$$\times \Big[\frac{\beta L c_m}{\epsilon(1-\beta)^2} + \frac{\rho}{\eta\epsilon^2}\frac{8\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3}r^3 c_m^3\big)}{(1-\beta)^3} + \frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon(1-\beta)}\Big]. \tag{4.63}$$

To show that the above is nonnegative, it suffices to show that

$$r^2\gamma \geq \frac{24\eta r\beta L c_m^2}{\epsilon(1-\beta)^2}, \tag{4.64}$$

and

$$r^2\gamma \geq \frac{24\eta r c_m \rho}{(1-\beta)\eta\epsilon^2}\frac{8\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3}r^3 c_m^3\big)}{(1-\beta)^2}, \tag{4.65}$$

and

$$r^2\gamma \geq \frac{24\eta r c_m}{1-\beta}\frac{\rho\big(8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2\big)}{2\eta\epsilon}. \tag{4.66}$$

Now w.l.o.g, we assume that c_m, L, σ^2, c', and ρ are not less than one and that $\epsilon \leq 1$. By using the values of parameters on Table 4.2, we have the following results; a sufficient condition of (4.64) is that

$$\frac{c_r}{c_\eta} \geq \frac{24 L c_m^2 \epsilon^2}{(1-\beta)^2}. \tag{4.67}$$

A sufficient condition of (4.65) is that

$$\frac{c_r}{c_F} \geq \frac{576 c_m \rho}{(1-\beta)^3}, \tag{4.68}$$

and

$$1 \geq \frac{1152 c_m \rho c_h c_r}{(1-\beta)^3}, \tag{4.69}$$

and

$$1 \geq \frac{192 c_m^4 \rho^2 c_r^2}{(1-\beta)^3}. \tag{4.70}$$

A sufficient condition of (4.66) is that

$$1 \geq \frac{96 c_m \rho (\sigma^2 + 3c_m^2) c_r \epsilon}{(1-\beta)^3}, \tag{4.71}$$

and a sufficient condition for the above (4.71), by the assumption that both $\sigma^2 \geq 1$ and $c_m \geq 1$, is

$$1 \geq \frac{576 c_m^3 \rho \sigma^2 c_r \epsilon}{(1-\beta)^3}. \tag{4.72}$$

Now let us verify if (4.67), (4.68), (4.69), (4.70), (4.72) are satisfied. For (4.67), using the constraint of c_η on Table 4.2, we have that $\frac{1}{c_\eta} \geq \frac{c_m^5 \rho L^2 \sigma^2 c' c_h}{c_1}$. Using this inequality, it suffices to let $c_r \geq \frac{c_0 \epsilon^2}{c_m^3 \rho L \sigma^2 c' c_h (1-\beta)^2}$ for getting (4.67), which holds by using the constraint that $c'(1-\beta)^2 > 1$ and $\epsilon \leq 1$. For (4.68), using the constraint of c_F on Table 4.2, we have that $\frac{1}{c_F} \geq \frac{c_m^4 \rho^2 L \sigma^4 c_h}{c_2}$. Using this inequality, it suffices to let $c_r \geq \frac{c_0}{c_m^3 \rho L \sigma^4 (1-\beta)^3}$, which holds by using the constraint that $\sigma^2(1-\beta)^3 > 1$. For (4.69), it needs $\frac{(1-\beta)^3}{1152 c_m \rho c_h} \geq \frac{c_0}{c_m^3 \rho L \sigma^2 c_h} \geq c_r$, which hold by using the constraint that $\sigma^2(1-\beta)^3 > 1$. For (4.70), it suffices to let $\frac{(1-\beta)^2}{14 c_m^2 \rho} \geq \frac{c_0}{c_m^3 \rho L \sigma^2 c_h} \geq c_r$ which holds by using the constraint that $\sigma^2(1-\beta)^3 > 1$. For (4.72), it suffices to let $\frac{(1-\beta)^3}{576 c_m^3 \rho \sigma^2 \epsilon} \geq \frac{c_0}{c_m^3 \rho L \sigma^2 c_h} \geq c_r$, which holds by using the constraint that $L(1-\beta)^3 > 1$ and $\epsilon \leq 1$. Therefore, by choosing the parameter values as Table 4.2, we can guarantee that $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta \mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{q,\mathcal{T}_{thred}-1}\rangle] \geq 0$.

Proving that $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta \mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{w,\mathcal{T}_{thred}-1}\rangle] \geq 0$: By Lemma 39 and Lemma 49, we have that

$$\begin{aligned} &\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] + 2\eta \mathbb{E}_{t_0}[\langle q_{v,\mathcal{T}_{thred}-1}, q_{w,\mathcal{T}_{thred}-1}\rangle] \\ &\geq \frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda)\big)^2 r^2 \gamma - \frac{2\eta r c'}{(1-\beta)}(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda))^2 \epsilon. \end{aligned} \tag{4.73}$$

To show that the above is nonnegative, it suffices to show that

$$r^2\gamma \geq \frac{8\eta r c' \epsilon}{(1-\beta)}. \tag{4.74}$$

A sufficient condition is $\frac{c_r}{c_\eta} \geq \frac{8\epsilon^4 c'}{1-\beta}$. Using the constraint of c_η on Table 4.2, we have that $\frac{1}{c_\eta} \geq \frac{c_m^5 \rho L^2 \sigma^2 c' c_h}{c_1}$. So, it suffices to let $c_r \geq \frac{c_0 \epsilon^4}{3 c_m^5 \rho L^2 \sigma^2 c_h (1-\beta)}$, which holds by using the constraint that $L(1-\beta)^3 > 1$ (so that $L(1-\beta) > 1$) and $\epsilon \leq 1$.

Proving that $\frac{1}{4}\mathbb{E}_{t_0}[\|q_{v,\mathcal{T}_{thred}-1}\|^2] \geq C_{upper}$:

From Lemma 39 and Lemma 36, we need to show that

$$\begin{aligned} &\frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda)\big)^2 r^2\gamma \\ &\geq \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2. \end{aligned} \tag{4.75}$$

We know that $\frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\lambda)\big)^2 r^2\gamma \geq \frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\epsilon)\big)^2 r^2\gamma$. It suffices to show that

$$\begin{aligned} &\frac{1}{4}\big(\Pi_{j=1}^{\mathcal{T}_{thred}-1}(1+\eta\theta_j\epsilon)\big)^2 r^2\gamma \\ &\geq \frac{8\eta t\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2} + 8\frac{r^2\sigma^2}{(1-\beta)^2} + 4\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + 2r^2 c_m^2. \end{aligned} \tag{4.76}$$

Note that the left hand side is exponentially growing in $\mathcal{T}_{thred}$. We can choose the number of iterations $\mathcal{T}_{thred}$ large enough to get the desired result. Specifically, we claim that $\mathcal{T}_{thred} \geq \frac{c(1-\beta)}{\eta\epsilon}\log(\frac{L c_m \sigma^2 \rho c' c_h}{(1-\beta)\delta\gamma\epsilon})$ for some constant $c > 0$. To see this, let us first apply $\log$ on both sides of (4.76),

$$2\big(\sum_{j=1}^{\mathcal{T}_{thred}-1}\log(1+\eta\theta_j\epsilon)\big) + \log(r^2\gamma) \geq \log(8a\mathcal{T}_{thred} + 8b) \tag{4.77}$$

where we denote $a := \frac{4\eta\big(\mathcal{F}_{thred} + 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3\big)}{(1-\beta)^2}$ and $b := 4\frac{r^2\sigma^2}{(1-\beta)^2} + 2\eta^2\big(\frac{\beta}{1-\beta}\big)^2 c_m^2 + r^2 c_m^2$. To proceed, we are going to use the inequality $\log(1+x) \geq \frac{x}{2}$, for $x \in [0, \sim 2.51]$. We have that

$$1 \geq \frac{\eta\epsilon}{(1-\beta)} \tag{4.78}$$

as guaranteed by the constraint of η. So,

$$\begin{aligned}
&2\Big(\sum_{j=1}^{\mathcal{T}_{thred}-1} \log(1+\eta\theta_j\epsilon)\Big) \overset{(a)}{\geq} \sum_{j=1}^{\mathcal{T}_{thred}-1} \eta\theta_j\epsilon = \sum_{j=1}^{\mathcal{T}_{thred}-1}\sum_{k=0}^{j-1} \beta^k\eta\epsilon \\
&= \sum_{j=1}^{\mathcal{T}_{thred}-1} \frac{1-\beta^j}{1-\beta}\eta\epsilon \geq \frac{1}{1-\beta}(\mathcal{T}_{thred} - 1 - \frac{\beta}{1-\beta})\eta\epsilon. \\
&\overset{(b)}{\geq} \frac{\mathcal{T}_{thred}-1}{2(1-\beta)}\eta\epsilon,
\end{aligned} \tag{4.79}$$

where (a) is by using the inequality $\log(1+x) \geq \frac{x}{2}$ with $x = \eta\theta_j\epsilon \leq 1$ and (b) is by making $\frac{\mathcal{T}_{thred}-1}{2(1-\beta)} \geq \frac{\beta}{(1-\beta)^2}$, which is equivalent to the condition that

$$\mathcal{T}_{thred} \geq 1 + \frac{2\beta}{1-\beta} \tag{4.80}$$

Now let us substitute the result of (4.79) back to (4.77). We have that

$$\mathcal{T}_{thred} \geq 1 + \frac{2(1-\beta)}{\eta\epsilon} \log(\frac{8a\mathcal{T}_{thred} + 8b}{\gamma r^2}), \tag{4.81}$$

which is what we need to show. By choosing $\mathcal{T}_{thred}$ large enough,

$$\mathcal{T}_{thred} \geq \frac{c(1-\beta)}{\eta\epsilon} \log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon}) = O((1-\beta)\log(\frac{1}{(1-\beta)\epsilon})\epsilon^{-6}) \tag{4.82}$$

for some constant $c > 0$, we can guarantee that the above inequality (4.81) holds.

4.4.5 Proof of Lemma 50

Lemma 50 ([63]) *Let us define the event* $\Upsilon_k := \{\|\nabla f(w_{k\mathcal{T}_{thred}})\| \geq \epsilon$ *or* $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \leq -\epsilon\}$. *The complement is* $\Upsilon_k^c := \{\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon$ *and* $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \geq -\epsilon\}$, *which suggests that $w_{k\mathcal{T}_{thred}}$ is an (ϵ, ϵ)-second order stationary points. Suppose that*

$$\begin{aligned}
\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k] \le -\Delta \\
\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k^c] \le \delta\frac{\Delta}{2}.
\end{aligned} \tag{4.83}$$

Set $T = 2\mathcal{T}_{thred}\big(f(w_0) - \min_w f(w)\big)/(\delta\Delta)$. We return w uniformly randomly from $w_0, w_{\mathcal{T}_{thred}}, w_{2\mathcal{T}_{thred}}, \dots, w_{k\mathcal{T}_{thred}}, \dots, w_{K\mathcal{T}_{thred}}$, where $K := \lfloor T/\mathcal{T}_{thred} \rfloor$. Then, with probability at least $1-\delta$, we will have chosen a w_k where Υ_k did not occur.

Proof. Let P_k be the probability that Υ_k occurs.

$$\begin{aligned}
&\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})] \\
&= \mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k]P_k + \mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k^c](1-P_k) \\
&\le -\Delta P_k + \delta\Delta/2(1-P_k) \\
&= \delta\Delta/2 - (1+\delta/2)\Delta P_k \\
&\le \delta\Delta/2 - \Delta P_k.
\end{aligned} \tag{4.84}$$

Summing over all K, we have

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})] \le \Delta\frac{1}{K+1}\sum_{k=0}^{K}(\delta/2 - P_k) \\
&\Rightarrow \frac{1}{K+1}\sum_{k=0}^{K}P_k \le \delta/2 + \frac{f(w_0) - \min_w f(w)}{(K+1)\Delta} \le \delta \\
&\Rightarrow \frac{1}{K+1}\sum_{k=0}^{K}(1-P_k) \ge 1-\delta.
\end{aligned} \tag{4.85}$$

□

4.4.6 Proof of Theorem 29

Theorem 29 *Assume that the stochastic momentum satisfies CNC. Set* $r = O(\epsilon^2)$*,* $\eta = O(\epsilon^5)$*, and* $\mathcal{T}_{thred} = \frac{c(1-\beta)}{\eta\epsilon} \log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon}) = O((1-\beta)\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon})\epsilon^{-6})$ *for some constant* $c > 0$*. If SGD with momentum (Algorithm 20) has APAG property when gradient is large (*$\|\nabla f(w)\| \geq \epsilon$*), APCG*$_{\mathcal{T}_{thred}}$ *property when it enters a region of saddle points that exhibits a negative curvature (*$\|\nabla f(w)\| \leq \epsilon$ *and* $\lambda_{\min}(\nabla^2 f(w)) \leq -\epsilon$*), and GrACE property throughout the iterations, then it reaches an* (ϵ, ϵ) *second order stationary point in* $T = 2\mathcal{T}_{thred}(f(w_0) - \min_w f(w))/(\delta \mathcal{F}_{thred}) = O((1-\beta)\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon})\epsilon^{-10})$ *iterations with high probability* $1-\delta$*, where* $\mathcal{F}_{thred} = O(\epsilon^4)$*.*

Proof sketch of Theorem 29 In this subsection, we provide a sketch of the proof of Theorem 29. The complete proof is available in Section 4.4.6. Our proof uses a lemma in ([63]), which is Lemma 50 below. The lemma guarantees that uniformly sampling a w from $\{w_{k\mathcal{T}_{thred}}\}$, $k = 0, 1, 2, \dots, \lfloor T/\mathcal{T}_{thred} \rfloor$ gives an (ϵ, ϵ)-second order stationary point with high probability. We replicate the proof of Lemma 50 in Section 4.4.5.

Lemma 50. *([63]) Let us define the event* $\Upsilon_k := \{\|\nabla f(w_{k\mathcal{T}_{thred}})\| \geq \epsilon \text{ or } \lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \leq -\epsilon\}$*. The complement is* $\Upsilon_k^c := \{\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon \text{ and } \lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \geq -\epsilon\}$*, which suggests that* $w_{k\mathcal{T}_{thred}}$ *is an* (ϵ, ϵ)*-second order stationary points. Suppose that*

$$\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}}) | \Upsilon_k] \leq -\Delta \quad \& \quad \mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}}) | \Upsilon_k^c] \leq \delta \frac{\Delta}{2}. \tag{4.86}$$

Set $T = 2\mathcal{T}_{thred}\big(f(w_0) - \min_w f(w)\big)/(\delta\Delta)$*.* [7] *We return* w *uniformly randomly from* $w_0, w_{\mathcal{T}_{thred}}, w_{2\mathcal{T}_{thred}}, \dots, w_{k\mathcal{T}_{thred}}, \dots, w_{K\mathcal{T}_{thred}}$*, where* $K := \lfloor T/\mathcal{T}_{thred} \rfloor$*. Then, with probability at least* $1-\delta$*, we will have chosen a* w_k *where* Υ_k *did not occur.*

To use the result of Lemma 50, we need to let the conditions in (4.86) be satisfied. We can bound $\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}}) | \Upsilon_k] \leq -\mathcal{F}_{thred}$, based on the analysis of the

[7] One can use any upper bound of $f(w_0) - \min_w f(w)$ as $f(w_0) - \min_w f(w)$ in the expression of T.

large gradient norm regime (Lemma 42) and the analysis for the scenario when the update is with small gradient norm but a large negative curvature is available (Subsection 4.3.3). For the other condition, $\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k^c] \leq \delta\frac{\mathcal{F}_{thred}}{2}$, it requires that the expected amortized increase of function value due to taking the large step size r is limited (i.e. bounded by $\delta\frac{\mathcal{F}_{thred}}{2}$) when $w_{k\mathcal{T}_{thred}}$ is a second order stationary point. By having the conditions satisfied, we can apply Lemma 50 and finish the proof of the theorem.

Proof of Theorem 29

Proof. Our proof is based on Lemma 50. So, let us consider the events in Lemma 50, $\Upsilon_k := \{\|\nabla f(w_{k\mathcal{T}_{thred}})\| \geq \epsilon$ or $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \leq -\epsilon\}$. We first show that $\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k] \leq \mathcal{F}_{thred}$.

When $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \geq \epsilon$:

Consider that Υ_k is the case that $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \geq \epsilon$. Denote $t_0 := k\mathcal{T}_{thred}$ in the following. We have that

$$
\begin{aligned}
&\mathbb{E}_{t_0}[f(w_{t_0+\mathcal{T}_{thred}}) - f(w_{t_0})] \\
&= \sum_{t=0}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]] \\
&= \mathbb{E}_{t_0}[f(w_{t_0+1}) - f(w_{t_0})] + \sum_{t=1}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]],
\end{aligned}
\tag{4.87}
$$

which can be further bounded as

$$
\begin{aligned}
&\overset{(a)}{\leq} -\frac{r}{2}\|\nabla f(w_{t_0})\|^2 + \frac{Lr^2c_m^2}{2} + \sum_{t=1}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]] \\
&\overset{(b)}{\leq} -\frac{r}{2}\|\nabla f(w_{t_0})\|^2 + \frac{Lr^2c_m^2}{2} + \sum_{t=1}^{\mathcal{T}_{thred}-1} \left(\eta^2 c_h + \frac{\rho}{6}\eta^3 c_m^3\right) \\
&\overset{(c)}{\leq} -\frac{r}{2}\|\nabla f(w_{t_0})\|^2 + \frac{Lr^2c_m^2}{2} + r^2 c_h + \frac{\rho}{6} r^3 c_m^3 \\
&\overset{(d)}{\leq} -\frac{r}{2}\|\nabla f(w_{t_0})\|^2 + Lr^2c_m^2 + r^2 c_h \\
&\overset{(e)}{\leq} -\frac{r}{2}\epsilon^2 + Lr^2c_m^2 + r^2 c_h \overset{(f)}{\leq} -\frac{r}{4}\epsilon^2 \overset{(g)}{\leq} -\mathcal{F}_{thred},
\end{aligned}
\tag{4.88}
$$

where (a) is by using Lemma 41 with step size r, (b) is by using Lemma 43, (c) is due to the constraint that $\eta^2 \mathcal{T}_{thred} \leq r^2$, (d) is by the choice of r, (e) is by $\|\nabla f(w_t)\| \geq \epsilon$, (f) is by the choice of r so that $r \leq \frac{\epsilon^2}{4(Lc_m^2 + c_h)}$, and (g) is by

$$
\frac{r}{4}\epsilon^2 \geq \mathcal{F}_{thred}. \tag{4.89}
$$

When $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \leq -\epsilon$:

The scenario that Υ_k is the case that $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \leq -\epsilon$ has been analyzed in Section 4.4.4, which guarantees that $\mathbb{E}[f(w_{t_0+\mathcal{T}_{thred}}) - f(w_{t_0})] \leq -\mathcal{F}_{thred}$ under the setting.

When $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \geq -\epsilon$:

Now let us switch to show that $\mathbb{E}[f(w_{(k+1)\mathcal{T}_{thred}}) - f(w_{k\mathcal{T}_{thred}})|\Upsilon_k^c] \leq \delta\frac{\mathcal{F}_{thred}}{2}$. Recall that Υ_k^c means that $\|\nabla f(w_{k\mathcal{T}_{thred}})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(w_{k\mathcal{T}_{thred}})) \geq -\epsilon$. Denote $t_0 :=$

$k\mathcal{T}_{thred}$ in the following. We have that

$$\begin{aligned}
&\mathbb{E}_{t_0}[f(w_{t_0+\mathcal{T}_{thred}}) - f(w_{t_0})] = \sum_{t=0}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]] \\
&= \mathbb{E}_{t_0}[f(w_{t_0+1}) - f(w_{t_0})] + \sum_{t=1}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]] \\
&\overset{(a)}{\leq} r^2 c_h + \frac{\rho}{6} r^3 c_m^3 + \sum_{t=1}^{\mathcal{T}_{thred}-1} \mathbb{E}_{t_0}[\mathbb{E}[f(w_{t_0+t+1}) - f(w_{t_0+t})|w_{0:t_0+t}]] \\
&\overset{(b)}{\leq} r^2 c_h + \frac{\rho}{6} r^3 c_m^3 + \sum_{t=1}^{\mathcal{T}_{thred}-1} \left(\eta^2 c_h + \frac{\rho}{6}\eta^3 c_m^3\right) \\
&\overset{(c)}{\leq} 2r^2 c_h + \frac{\rho}{3} r^3 c_m^3 \leq 4r^2 c_h \overset{(d)}{\leq} \frac{\delta \mathcal{F}_{thred}}{2}.
\end{aligned} \tag{4.90}$$

where (a) is by using Lemma 43 with step size r, (b) is by using Lemma 43 with step step size η, (c) is by setting $\eta^2 \mathcal{T}_{thred} \leq r^2$ and $\eta \leq r$, (d) is by the choice of r so that $8r^2 c_h \leq \delta \mathcal{F}_{thred}$.

Now we are ready to use Lemma 50, since both the conditions are satisfied. According to the lemma and the choices of parameters value on Table 4.2, we can set $T = 2\mathcal{T}_{thred}\big(f(w_0) - \min_w f(w)\big)/(\delta \mathcal{F}_{thred}) = O((1-\beta)\log(\frac{Lc_m\sigma^2\rho c' c_h}{(1-\beta)\delta\gamma\epsilon})\epsilon^{-10})$, which will return a w that is an (ϵ, ϵ) second order stationary point. Thus, we have completed the proof.

□

4.5 Discussion: Over-parametrization

In the previous sections, we show that Polyak's momentum helps fast saddle point escape. In this section, we consider a different technique — over-parametrization, which is recently very popular in modern machine learning. Specifically, let us consider over-parametrizing the phase retrieval problem (4.4) as follows,

$$\min_{W \in \mathbb{R}^{d \times K}} \tfrac{1}{4n} \textstyle\sum_{i=1}^{n} \big((x_i^\top w^{(1)})^2 + (x_i^\top w^{(2)})^2 + ... + (x_i^\top w^{(K)})^2 - y_i\big)^2, \tag{4.91}$$

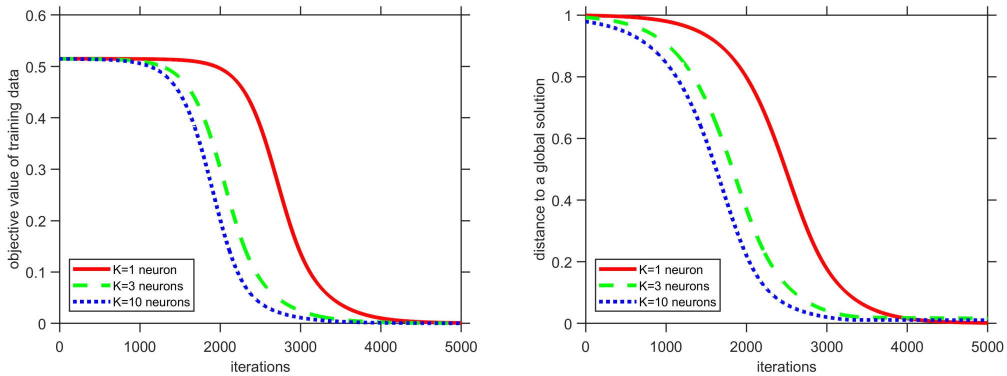
where $x_i \sim N(0, I_d)$, $y_i = (x_i^\top w_*)^2$, and $w_* \in \mathbb{R}^d$. This is over-parametrization since the objective has more variables than necessary.

Let us try a simulation. We set the dimension $d = 10$ and the number of training samples $n = 200$. We let $w_* = e_1$ with e_1 being the unit vector. Each neuron $w^{(k)} \in \mathbb{R}^d$ ($k \in [K]$) of the student network is initialized by sampling from an isotropic distribution and is close to the origin (i.e. $w_0^{(k)} \sim 0.01 \cdot N(0, I_d/d)$). Figure 4.5 show a very interesting result of applying vanilla gradient descent to train different sizes of models. Each curve represents the progress of gradient descent for different K. It shows that for a larger K, gradient descent escapes "the origin" faster, however, we remark that the origin is different for a different K, as each problem of K has a different dimensional parameter space.

We also report a *distance* measure on the same figure.

$$\text{dist}(W, w_*) := \min_{q \in \mathbb{R}^K : \|q\|_2 \le 1} \|W - w_* q^\top\|. \tag{4.92}$$

This is due to our observation that for any K, the global optimal solutions of (4.91) that achieve zero testing error are $w_* q^\top \in \mathbb{R}^{d \times K}$ for any $q \in \mathbb{R}^K$ such that $\|q\|_2 = 1$. To see this, substitute $W = w_* q^\top \in \mathbb{R}^{d \times K}$ into (4.91). We have that for any $x_i \in \mathbb{R}^d$ it holds that $(x_i^\top w^{(1)})^2 + (x_i^\top w^{(2)})^2 + ... + (x_i^\top w^{(K)})^2 - y_i = \|x_i^\top W\|_F^2 - (x_i^\top w_*)^2 = \text{tr}\big((x_i^\top w_* q^\top)^\top (x_i^\top w_* q^\top)\big) - (x_i^\top w_*)^2 = 0$. Therefore, the metric $\text{dist}(W, w_*)$ as be viewed as a surrogate of the testing error. In particular, $\text{dist}(W_t, w_*)$ represents the distance of the current iterate W_t and its closest global optimal solution to the over-parametrized objective (4.91) that achieves zero testing error. Note that the argmin of (4.92) is $q_* := \frac{W^\top w_*}{\|W^\top w_*\|_2} = \arg\min_{q \in \mathbb{R}^K : \|q\|_2 \le 1} \|W - w_* q^\top\|$. Subfigure (b) of Figure 4.5 plots the distance of the iterates generated by gradient descent and its closet global optimal solution for different sizes K of models. We see that over-parametrization enables shrinking the distance $\text{dist}(W_t^{\#K}, w_*)$ faster.



(a) Objective value (4.91) vs. iteration t. (b) Distance (4.92) vs. iteration t.

Figure 4.5: Vanilla gradient descent for training different over-parametrized models (4.91). We see that over-parametrization helps the iterate of gradient descent escapes the origin faster and hence converges faster.

Informal analysis Let us provide an informal analysis to explain why over-parametrization might help the fast escape. Given the infinite number of samples $x_i \sim N(0, I_d)$, the population objective of (4.91) is

$$F(W) := \left(\sum_{k=1}^{K} \|w^{(k)}\|^2 - \|w_*\|^2 \right)^2 + 2\| \sum_{k=1}^{K} w^{(k)}(w^{(k)})^\top - w_* w_*^\top \|_F^2. \tag{4.93}$$

The Hessian at the origin $\nabla^2 F(0_{d\times K}) \in \mathbb{R}^{dK\times dK}$ is in the following form,

$$\begin{bmatrix} -4\|w_*\|I_d - 8w_*w_*^\top & & & \\ & -4\|w_*\|I_d - 8w_*w_*^\top & & \\ & & \ddots & \\ & & & -4\|w_*\|I_d - 8w_*w_*^\top \end{bmatrix}.$$

Let $v^{(K)} \in \mathbb{R}^{d\times K}$ be the bottom eigenvector of $\nabla^2 F(0_{d\times K})$ and $v^{(1)} \in \mathbb{R}^d$ be the bottom eigenvector of $\nabla^2 F(0_{d\times 1})$. We might be able to write

$$(v^{(K)})^\top \nabla^2 F(0_{d\times K}) v^{(K)} = K \times (v^{(1)})^\top \nabla^2 F(0_{d\times 1}) v^{(1)}. \tag{4.94}$$

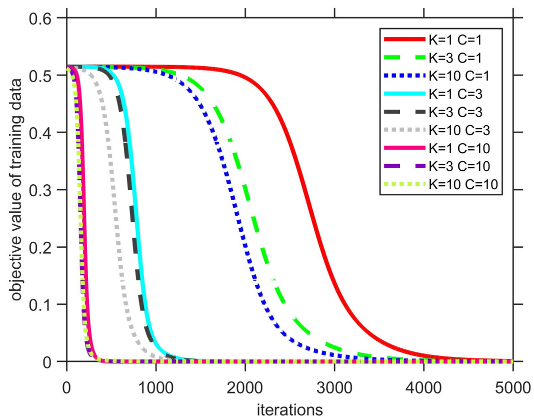


Figure 4.6: Gradient descent for (4.95) under different sizes K and scales of the outputs C

That is, effectively a K-times-larger size of models results in a k-times-larger negative curvature at the origin.

A natural question is then "*Is the effect of over-parametrization equivalent to using a larger step size* η*?*". To answer the question, let us also consider an over-parametrized version of (4.95),

$$\min_{W \in \mathbb{R}^{d \times K}} \hat{f}(W) := \frac{1}{4n} \sum_{i=1}^{n} \left(C(x_i^\top w^{(1)})^2 + C(x_i^\top w^{(2)})^2 + \dots + C(x_i^\top w^{(K)})^2 - y_i \right)^2. \tag{4.95}$$

Applying vanilla gradient descent to (4.95) of $K = 1$ might be viewed as using a C-times larger step size as if it were optimizing (4.91), since $\nabla \hat{f}(W) = \frac{C}{n} \sum_{i=1}^{n} \left(C(x_i^\top w^{(1)})^2 - y_i \right) x_i$. On Figure 4.6, we report gradient descent with the same step size η for solving (4.95) under different C's and K's. It suggests that *to some degree*, over-parametrization is kind of like using a larger step size. However, in order to converge to a good solution, an upper-bound of the step size should be required. A deeper investigation needs to be conducted. It is also interesting to check if the effect of over-parametrization also exists in other problems as well, not necessarily limited to phase retrieval.

Related works The observation that a larger network can be trained to achieve a certain level of prediction performance with fewer iterations than that of a smaller net can be dated back as early as the work of Livni et al. (Section 5 of [182]), who try different levels of over-parametrization and report that SGD converges much faster and finds a better solution when it is used to train a larger network. However, the reason why over-parametrization can lead to an acceleration still remains a mystery, and very little theory has helped explain the observation, with perhaps the notable exception of Arora et al. [15]. Arora et al. [15] consider over-parametrizing a single-output linear regression with l_p loss for $p > 2$– the square loss corresponds to $p = 2$–and they study the linear regression problem by replacing the model $w \in \mathbb{R}^d$ by another model $w_1 \in \mathbb{R}^d$ times a scalar $w_2 \in \mathbb{R}$. They show that the dynamics of gradient descent on the new over-parametrized model are equivalent to the dynamics of gradient descent on the original objective function with an adaptive learning rate plus some momentum terms. However, in practice, people actually use the techniques of over-parametrization, adaptive learning rate, and momentum simultaneously in deep learning (see e.g. [131, 151, 185, 252]), as each technique appears to contribute to performance and they may, to some extent, be complementary. It has been suggested that over-parameterizing a model leads implicitly to an adaptive learning rate or momentum, but this does not appear to fully explain the performance improvement.

Finally, we also want to acknowledge some related works of understanding over-parametrization in different aspects (e.g. [36, 83, 202]).

4.6 Conclusion

In this work, we identify three properties that guarantee SGD with momentum in reaching a second-order stationary point faster by a higher momentum, which justifies the practice of using a large value of momentum parameter β. We show that a greater momentum leads to escaping strict saddle points faster due to that SGD with momentum recursively enlarges the projection to an escape direction. However, how to make sure that SGD with momentum has the three properties is not very clear. It would be interesting to identify conditions that guarantee SGD with momentum to have the properties. Perhaps a good starting point is understanding why the properties hold in phase retrieval. We also discuss the effect of over-parametrization and report some interesting observations. We hope our results shed light on understanding the interaction between momentum and over-parametrization for exploiting negative curvatures.

REFERENCES

[1] Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. "Online linear optimization via smoothing". In: *COLT* (2014).

[2] Jacob Abernethy, Kfir Levy, Kevin Lai, and Jun-Kun Wang. "Faster Rates for Convex-Concave Games". In: *COLT* (2018).

[3] Jacob Abernethy and Jun-Kun Wang. "On Frank-Wolfe and Equilibrium Computation". In: *NeurIPS* (2017).

[4] Jacob Abernethy, Manfred K Warmuth, and Joel Yellin. "Optimal strategies from random walks". In: *COLT* (2008).

[5] Ilan Adler. "The equivalence of linear programs and zero-sum games". In: *International Journal of Game Theory* 42.1 (Feb. 2013), pp. 165–177.

[6] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. "Finding approximate local minima faster than gradient descent". In: *STOC* (2017).

[7] Ahmet Alacaoglu, Yura Malitsky, Panayotis Mertikopoulos, and Volkan Cevher. "A new regret analysis for Adam-type algorithms". In: *ICML* (2020).

[8] Zeyuan Allen-Zhu, Elad Hazan, Wei Hu, and Yuanzhi Li. "Linear Convergence of a Frank-Wolfe Type Algorithm over Trace-Norm Balls". In: *NeurIPS* (2017).

[9] Zeyuan Allen-Zhu and Yuanzhi Li. "Neon2: Finding local minima via first-order oracles". In: *NeurIPS* (2018).

[10] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. "A convergence theory for deep learning via overparameterization". In: *ICML* (2019).

[11] Zeyuan Allen-Zhu and Lorenzo Orecchia. "Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent". In: *ITCS* (2017).

[12] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, and et al. "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin". In: *ICML* (2016).

[13] Anima Anandkumar and Rong Ge. "Efficient approaches for escaping higher order saddle points in non-convex optimization". In: *COLT* (2016).

[14] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks". In: *ICLR* (2019).

[15] Sanjeev Arora, Nadav Cohen, and Elad Hazan. “On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization”. In: *ICML* (2018).

[16] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. “Implicit Regularization in Deep Matrix Factorization”. In: *NerurIPS* (2019).

[17] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. “Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks”. In: *NeurIPS* (2019).

[18] Sanjeev Arora, Elad Hazan, and Satyen Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications.” In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[19] Jean-Francois Aujol, Charles Dossal, and Aude Rondepierre. “Convergence rates of the Heavy-Ball method with Lojasiewicz property”. In: *hal-02928958* (2020).

[20] Francis Bach. “Duality between subgradient and conditional gradient methods”. In: *SIAM Journal on Optimization* (2015).

[21] Yu Bai and Jason D. Lee. “Beyond Linearization: On Quadratic and Higher-Order Approximation of Wide Neural Networks”. In: *ICLR* (2020).

[22] David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. “The Mechanics of n-Player Differentiable Games”. In: *ICML* (2018).

[23] Heinz H. Bauschke and Yves Lucet. “What is a Fenchel Conjugate?” In: *Notices of the AMS* (2012).

[24] Amir Beck and Shimrit Shtern. “Linearly convergent away-step conditional gradient for non-strongly convex functions”. In: *Mathematical Programming* (2017).

[25] Amir Beck and Marc Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM J. on Imaging Sciences* (2009).

[26] Aharon Ben-Tal and Arkadi Nemirovski. “Lectures on Modern Convex Optimization”. In: *MPS-SIAM Series on Optimization, SIAM* (2001).

[27] Dimitri P. Bertsekas, Angelia Nedic, and Asuman E. Ozdaglar. “Convex Analysis and Optimization”. In: *Athena Scientific* (2003).

[28] Alberto Bietti and Julien Mairal. “On the Inductive Bias of Neural Tangent Kernels”. In: *NeurIPS* (2019).

[29] Jonathan Borwein and Adrian S Lewis. "Convex Analysis and Nonlinear Optimization Theory and Examples". In: *Springer* (2006).

[30] Stephen Boyd and Lieven Vandenberghe. "Convex Optimization". In: *Cambridge University Press* (2004).

[31] Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. "Training (Overparametrized) Neural Networks in Near-Linear Time". In: *arXiv:2006.11648* (2020).

[32] Gábor Braun, Sebastian Pokutta, Dan Tu, and Stephen Wright. "Blended Conditonal Gradients". In: *ICML* (2019).

[33] Gabor Braun, Sebastian Pokutta, and Daniel Zink. "Lazifying Conditional Gradient Algorithms". In: *ICML* (2017).

[34] Gábor Braun, Sebastian Pokutta, and Daniel Zink. "Lazifying Conditional Gradient Algorithms". In: *JMLR* (2019).

[35] Alon Brutzkus and Amir Globerson. "Globally Optimal Gradient Descent for a ConvNet with Gaussian Inputs". In: *ICML* (2017).

[36] Alon Brutzkus and Amir Globerson. "Why do Larger Models Generalize Better? A Theoretical Perspective via the XOR Problem". In: *ICML* (2019).

[37] Sabastien Bubeck, Yin Tat Lee, and Mohit Singh. "A geometric alternative to Nesterov's accelerated gradient descent". In: *arXiv:1506.08187* (2015).

[38] Sebastien Bubeck. "Convex Optimization: Algorithms and Complexity". In: *Foundations and Trends in Machine Learning* (2015).

[39] Tianle Cai, Ruiqi Gao, Jikai Hou, Siyu Chen, Dong Wang, Di He, Zhihua Zhang, and Liwei Wang. "A gram-gauss-newton method learning overparameterized deep neural networks for regression problems". In: *arXiv.org:1905.11675* (2019).

[40] Bugra Can, Mert Gürbüzbalaban, and Lingjiong Zhu. "Accelerated Linear Convergence of Stochastic Momentum Methods in Wasserstein Distances". In: *ICML* (2019).

[41] Emmanuel J. Candés, Yonina Eldar, Thomas Strohmer, and Vlad Voroninski. "Phase retrieval via matrix completion." In: *SIAM Journal on Imaging Sciences* (2013).

[42] Emmanuel J Candes, Xiaodong Li, Yi Ma, and John Wright. "Robust principal component analysis?" In: *Journal of the ACM (JACM)* (2011).

[43] Emmanuel Candes and Benjamin Recht. “Exact matrix completion via convex optimization”. In: *Communications of the ACM* (2012).

[44] Yair Carmon and John C. Duchi. “Gradient Descent Efficiently Finds the Cubic-Regularized Non-Convex Newton Step”. In: *https://arxiv.org/pdf/1612.00547.pdf* (2017).

[45] Yair Carmon, John C. Duchi, Aaron Sidford, and Kevin Tian. “A Rank-1 Sketch for Matrix Multiplicative Weights”. In: *COLT* (2019).

[46] Yair Carmon, John Duchi, Oliver Hinder, and Aaron Sidford. “Accelerated Methods for NonConvex Optimization”. In: *SIAM Journal of Optimization* (2018).

[47] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. *On the Generalization Ability of On-Line Learning Algorithms*. IEEE Transactions on Information Theory, 2004.

[48] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[49] Gong Chen and Marc Teboulle. “Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions”. In: *SIAM Journal on Optimization* (1993).

[50] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. “Projection-free online optimization with stochastic gradient: From convexity to submodularity”. In: *ICML* (2018).

[51] Shuxiao Chen, Hangfeng He, and Weijie J. Su. “Label-Aware Neural Tangent Kernel: Toward Better Generalization and Local Elasticity”. In: *NeurIPS* (2020).

[52] Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, and Yuling Yan. “Gradient Descent with Random Initialization: Fast Global Convergence for Nonconvex Phase Retrieval”. In: *Mathematical Programming* (2018).

[53] Zixiang Chen, Yuan Cao, Quanquan Gu, and Tong Zhang. “A Generalized Neural Tangent Kernel Analysis for Two-layer Neural Network”. In: *NeurIPS* (2020).

[54] Lenaic Chizat, Edouard Oyallon, and Francis Bach. “On lazy training in differentiable programming”. In: *NeurIPS* (2019).

[55] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. “The Loss Surfaces of Multilayer Networks”. In: *AISTAT* (2015).

[56] Ken Clarkson. “Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm”. In: *SODA* (2008).

[57] Michael B. Cohen, Aaron Sidford, and Kevin Tian. “Relative Lipschitzness in Extragradient Methods and a Direct Recipe for Acceleration”. In: *arXiv:2011.06572* (2020).

[58] Cyrille W. Combettes and Sebastian Pokutta. “Boosting Frank-Wolfe by Chasing Gradients”. In: *ICML* (2020).

[59] Cyrille W. Combettes and Sebastian Pokutta. “Complexity of Linear Minimization and Projection on Some Sets”. In: *arXiv:2101.10040* (2021).

[60] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. “Autoaugment: Learning augmentation policies from data.” In: *arXiv:1805.09501* (2018).

[61] Frank E. Curtis, Daniel P. Robinson, and Mohammadreza Samadi. “A Trust Region Algorithm with a Worst-case Iteration Complexity of $O(\epsilon^{-3/2})$ for Nonconvex Optimization”. In: *Mathematical Programming* (2017).

[62] Ashok Cutkosky and Francesco Orabona. “Momentum-Based Variance Reduction in Non-Convex SGD”. In: *NeurIPS* (2019).

[63] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. “Escaping Saddles with Stochastic Gradients”. In: *ICML* (2018).

[64] Amit Daniely. “SGD Learns the Conjugate Kernel Class of the Network”. In: *NeurIPS* (2017).

[65] Amit Daniely. “Memorizing Gaussians with no over-parameterizaion via gradient decent on neural networks”. In: *arXiv:1909.11837* (2020).

[66] Marina Danilova, Anastasiya Kulakova, and Boris Polyak. “Non-monotone Behavior of the Heavy Ball Method”. In: *arXiv:1811.00658* (2018).

[67] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. “Training GANs with Optimism”. In: *arXiv preprint arXiv:1711.00141* (2017).

[68] Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”. In: *NIPS* (2014).

[69] Vladimir F. Demyanov and Aleksandr M. Rubinov. “Approximate methods in optimization problems”. In: *Elsevier Publishing Company,* (1970).

[70] Jelena Diakonikolas and Michael I. Jordan. "Generalized Momentum-Based Methods: A Hamiltonian Perspective". In: *arXiv:1906.00436* (2019).

[71] Jelena Diakonikolas and Lorenzo Orecchia. "The Approximate Duality Gap Technique: A Unified Theory of First-Order Methods". In: *SIAM Journal on Optimization* (2019).

[72] Jelena Diakonikolas1 and Lorenzo Orecchia. "Accelerated Extra-Gradient Descent: A Novel Accelerated First-Order Method". In: *ITCS* (2018).

[73] Lijun Ding, Yingjie Fei, Qiantong Xu, and Chengrun Yan. "Spectral Frank-Wolfe Algorithm: Strict Complementarity and Linear Convergence". In: *ICML* (2020).

[74] Simon S. Du and Wei Hu. "Width Provably Matters in Optimization for Deep Linear Neural Networks". In: *ICML* (2019).

[75] Simon S. Du, Chi Jin, Jason D. Lee, Michael I. Jordan, Barnabas Poczos, and Aarti Singh. "Gradient Descent Can Take Exponential Time to Escape Saddle Points". In: *NIPS* (2017).

[76] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. "Gradient Descent Provably Optimizes Over-parameterized Neural Networks". In: *ICLR* (2019).

[77] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. "Gradient descent finds global minima of deep neural networks". In: *ICML* (2019).

[78] John Duchi, Peter L. Bartlett, and Martin Wainwright. "Randomized smoothing for stochastic optimization". In: *SIAM Journal on Optimization (SIOPT)* (2012).

[79] Miroslav Dudik, Zaid Harchaoui, and Jerome Malick. "Lifted coordinate descent for learning with trace-norm regularization". In: *AISTAT* (2012).

[80] Yonatan Dukler, Quanquan Gu, and Guido Montufar. "Optimization Theory for ReLU Neural Networks Trained with Normalization Layers". In: *ICML* (2020).

[81] Joseph C. Dunn. "Rates of Convergence for Conditional Gradient Algorithms Near Singular and Nonsingular Extremals." In: *SIAM Journal on Control and Optimization* (1979).

[82] Pavel Dvurechensky, Petr Ostroukhov, Kamil Safin, Shimrit Shtern, and Mathias Staudigl. "Self-Concordant Analysis of Frank-Wolfe Algorithms". In: *ICML* (2020).

[83] Matt Emschwiller, David Gamarnik, Eren C. Kızıldag, and Ilias Zadik. “Neural Networks and Polynomial Regression. Demystifying the Overparametrization Phenomena”. In: *arXiv:2003.10523* (2020).

[84] Cong Fang, Hanze Dong, and Tong Zhang. “Over parameterized two-level neural networks can learn near optimal feature representations”. In: *arXiv:1910.11508* (2019).

[85] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. “Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator”. In: *NeurIPS* (2018).

[86] Cong Fang, Zhouchen Lin, and Tong Zhang. “Sharp Analysis for Nonconvex SGD Escaping from Saddle Points”. In: *COLT* (2019).

[87] Nicolas Flammarion and Francis Bach. “From Averaging to Acceleration, There is Only a Step-size”. In: *COLT* (2015).

[88] Nicolas Flammarion and Francis Bach. “From Averaging to Acceleration, There is Only a Step-size”. In: *COLT* (2015).

[89] Simon Foucart. “Matrix Norms and Spectral Radii”. In: *Online lecture note* (2018).

[90] Guilherme Franca, Jeremias Sulam, Daniel P Robinson, and Rene Vidal. “Conformal symplectic and relativistic optimization”. In: *Journal of Statistical Mechanics: Theory and Experiment* (2020).

[91] Marguerite Frank and Philip Wolfe. “An algorithm for quadratic programming”. In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110.

[92] Robert M. Freund. “Dual gauge programs, with applications to quadratic programming and the minimum-norm problem”. In: *Mathematical Programming* (1987).

[93] Robert M. Freund and Paul Grigas. “New analysis and results for the Frank–Wolfe method”. In: *Mathematical Programming* (2016).

[94] Yoav Freund and Robert E Schapire. “Game theory, on-line prediction and boosting”. In: *COLT* (1996), pp. 325–332.

[95] Yoav Freund and Robert E Schapire. “Adaptive game playing using multiplicative weights”. In: *Games and Economic Behavior* 29.1-2 (1999), pp. 79–103.

[96] Sébastien Gadat, Fabien Panloup, and Sofiane Saadane. “Stochastic Heavy Ball”. In: *arXiv:1609.04228* (2016).

[97] Dan Garber. “Fast projection-free convex optimization over the spectrahedron”. In: *NIPS* (2016).

[98] Dan Garber. “On the Convergence of Projected-Gradient Methods with Low-Rank Projections for Smooth Convex Minimization over Trace-Norm Balls and Related Problems”. In: *arXiv:1902.01644* (2019).

[99] Dan Garber and Elad Hazan. “Playing Non-linear Games with Linear Oracles.” In: *FOCS* (2013).

[100] Dan Garber and Elad Hazan. “Faster Rates for the Frank-Wolfe Method over Strongly-Convex Sets.” In: *ICML* (2015).

[101] Dan Garber and Elad Hazan. “A Linearly Convergent Conditional Gradient Algorithm With Applications to Online and Stochastic Optimization.” In: *SIAM Journal on Optimization* (2016).

[102] Dan Garber and Atara Kaplan. “On the Efficient Implementation of the Matrix Exponentiated Gradient Algorithm for Low-Rank Matrix Optimization”. In: *arXiv:2012.10469* (2020).

[103] Dan Garber and Ofer Meshi. “Linear-Memory and Decomposition-Invariant Linearly Convergent Conditional Gradient Algorithm for Structured Polytopes.” In: *NeurIPS* (2016).

[104] Xavier Gastaldi. “Shake-Shake regularization”. In: *arXiv:1705.07485* (2017).

[105] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. “Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition”. In: *COLT* (2015).

[106] Rong Ge, Rohith Kuditipudi, Zhize Li, and Xiang Wang. “Learning Two-layer Neural Networks with Symmetric Inputs”. In: *ICLR* (2019).

[107] I. Gelfand. “Normierte ringe”. In: *Mat. Sbornik* (1941).

[108] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. “Global convergence of the Heavy-ball method for convex optimization”. In: *ECC* (2015).

[109] Saeed Ghadimi and Guanghui Lan. “Stochastic first- and zeroth-order methods for nonconvex stochastic programming”. In: *SIAM Journal on Optimization* (2013).

[110] Saeed Ghadimi and Guanghui Lan. “Accelerated gradient methods for nonconvex nonlinear and stochastic programming”. In: *Mathematical Programming* (2016).

[111] G. Gidel, T. Jebara, and S. Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS* (2016).

[112] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. "Implicit Regularization of Discrete Gradient Dynamics in Linear Neural Networks". In: *NeurIPS* (2019).

[113] Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Gabriel Huang, Remi Lepriol, Simon Lacoste-Julien, and Ioannis Mitliagkas. "Negative momentum for improved game dynamics". In: *arXiv preprint arXiv:1807.04740* (2018).

[114] Gauthier Gidel, Fabian Pedregosa, and Simon Lacoste-Julien. "Frank-Wolfe Splitting via Augmented Lagrangian Method". In: *AISTATS* (2016).

[115] Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. "Understanding the Role of Momentum in Stochastic Gradient Methods". In: *NeurIPS* (2019).

[116] Gabriel Goh. "Why Momentum Really Works". In: *Distill* (2017).

[117] Gene H. Golub and Charles F. Van Loan. "Matrix computations". In: *Johns Hopkins University Press* (1996).

[118] Mehmet Gonen and Ethem Alpaydın. "Multiple kernel learning algorithms". In: *Journal of Machine Learning Research (JMLR)* (2011).

[119] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *NeurIPS* (2014), pp. 2672–2680.

[120] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. "Implicit Regularization in Matrix Factorization". In: *NeurIPS* (2017).

[121] Boris Hanin and Mihai Nica. "Finite Depth and Width Corrections to the Neural Tangent Kernel". In: *ICLR* (2020).

[122] James Hannan. "APPROXIMATION TO BAYES RISK IN REPEATED PLAY". In: *Contributions to the Theory of Games* 39 (1957), p. 97.

[123] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. "Conditional Gradient Algorithms for Norm-Regularized Smooth Convex Optimization". In: *Mathematical Programming* (2013).

[124] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. "Conditional gradient algorithms for Norm-regularized Smooth Convex Optimization". In: *Math. Prog., Series A* (2013).

[125] Moritz Hardt and Tengyu Ma. "Identity matters in deep learning". In: *ICLR* (2016).

[126] Elad Hazan. "Introduction to Online Convex Optimization". In: (2019).

[127] Elad Hazan et al. "Introduction to online convex optimization". In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325.

[128] Elad Hazan and Satyen Kale. "Projection-free Online Learning". In: *ICML* (2012).

[129] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[130] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. "Convex Analysis and Minimization Algorithms". In: *Springer* (1993).

[131] Elad Hoffer, Itay Hubara, and Daniel Soudry. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks". In: *NIPS* (2017).

[132] Bin Hu. "Unifying the Analysis in Control and Optimization via Semidefinite Programs". In: *Lecture Note* (2020).

[133] Bin Hu and Laurent Lessard. "Dissipativity Theory for Nesterov's Accelerated Method". In: *ICML* (2017).

[134] Wei Hu, Lechao Xiao, Ben Adlam, and Jeffrey Pennington. "The Surprising Simplicity of the Early-Time Learning Dynamics of Neural Networks". In: *NeurIPS* (2020).

[135] Wei Hu, Lechao Xiao, and Jeffrey Pennington. "Provable Benefit of Orthogonal Initialization in Optimizing Deep Linear Networks". In: *ICLR* (2020).

[136] Chien-Hao Huang, Jein-Shan Chen, and Chu-Chin Hu. "The Schatten p-norm on R^n". In: *Journal of Nonlinear and Convex Analysis* (2020).

[137] Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvari. "Following the Leader and Fast Rates in Linear Prediction: Curved Constraint Sets and Other Regularities". In: *NeurIPS* (2016).

[138] Arthur Jacot, Franck Gabriel, and Clement Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *NeurIPS* (2018).

[139] Martin Jaggi. "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization". In: *ICML* (2013).

[140] Martin Jaggi and Marek Sulovský. “A Simple Algorithm for Nuclear Norm Regularized Problems”. In: *ICML* (2010).

[141] Prateek Jain, Raghu Meka, and Inderjit S. Dhillon. “Guaranteed rank minimization via singular value projection”. In: *NeurIPS* (2010).

[142] Shuiwang Ji and Jieping Ye. “An accelerated gradient method for trace norm minimization”. In: *ICML* (2009).

[143] Ziwei Ji and Matus Telgarsky. “Gradient descent aligns the layers of deep linear networks”. In: *ICLR* (2019).

[144] Ziwei Ji and Matus Telgarsky. “Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks”. In: *ICLR* (2020).

[145] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. “How to Escape Saddle Points Efficiently”. In: *ICML* (2017).

[146] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. “Stochastic Gradient Descent Escapes Saddle Points Efficiently”. In: *arXiv:1902.04811* (2019).

[147] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. “Accelerated Gradient Descent Escapes Saddle Points Faster than Gradient Descent”. In: *COLT* (2018).

[148] Sham M. Kakade, Shai Shalev-shwartz, and Ambuj Tewari. “On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization”. In: *SIAM Journal on Control and Optimization* (2009).

[149] Adam Kalai and Santosh Vempala. “Efficient algorithms for online decision problems”. In: *Journal of Computer and System Sciences* 71.3 (2005), pp. 291–307.

[150] Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham M. Kakade. “On the insufficiency of existing momentum schemes for Stochastic Optimization”. In: *ICLR* (2018).

[151] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *ICLR* (2015).

[152] Jonas Moritz Kohler and Aurelien Lucchi. “Sub-sampled Cubic Regularization for Non-convex Optimization”. In: *ICML* (2017).

[153] Tomer Koren and Kfir Levy. “Fast rates for exp-concave empirical risk minimization”. In: *NeurIPS* (2015).

[154] Walid Krichene. “Continuous and Discrete Dynamics for Online Learning and Convex Optimization”. In: *Doctoral dissertation* (2016).

[155] Walid Krichene, Kenneth F. Caluyay, and Abhishek Halder. “Global Convergence of Second-order Dynamics in Two-layer Neural Networks”. In: *arXiv:2006.07867* (2020).

[156] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NIPS* (2012).

[157] Simon Lacoste-Julien and Martin Jaggi. “On the Global Linear Convergence of Frank-Wolfe Optimization Variants”. In: *NeurIPS* (2015).

[158] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. “Block-Coordinate Frank-Wolfe Optimization for Structural SVMs”. In: *ICML* (2013).

[159] Guanghui Lan. “The Complexity of Large-scale Convex Programming under a Linear Optimization Oracle”. In: *arXiv:1309.5550* (2013).

[160] Guanghui Lan. “First-order and Stochastic Optimization Methods for Machine Learning”. In: *Springer* (2020).

[161] Guanghui Lan and Yi Zhou. “Conditional gradient sliding for convex optimization”. In: *SIAM Journal on Optimization* (2016).

[162] Guanghui Lan and Yi Zhou. “An optimal randomized incremental gradient method”. In: *Mathematical programming* (2018).

[163] Thomas Laurent and James von Brecht. “Deep linear networks with arbitrary loss: All local minima are global”. In: *ICML* (2018).

[164] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. “Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent”. In: *NeurIPS* (2019).

[165] Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. “First-order methods almost always avoid strict saddle-points”. In: *Mathematical Programming, Series B* (2019).

[166] Jason D. Lee, Ruoqi Shen, Zhao Song, Mengdi Wang, and Zheng Yu. “Generalized Leverage Score Sampling for Neural Networks”. In: *arXiv:2009.09829* (2020).

[167] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I. Jordan. “Nonconvex finite-sum optimization via scsg methods”. In: *NIPS* (2017).

[168] Laurent Lessard, Benjamin Recht, and Andrew Packard. "Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints". In: *SIAM Journal on Optimization* (2016).

[169] Evgeny S Levitin and Boris T Polyak. "Constrained minimization methods." In: *USSR Computational mathematics and mathematical physics* (1966).

[170] Kfir Levy. "Online to Offline Conversions, Universality and Adaptive Minibatch Sizes". In: *NeurIPS* (2017).

[171] Kfir Y. Levy. "The power of normalization: Faster evasion of saddle points". In: *arXiv:1611.04831* (2016).

[172] Adrian S. Lewis and Hristo S. Sendov. "Twice differentiable spectral functions". In: *SIAM Journal on Matrix Analysis and Applications* (2001).

[173] Yuanzhi Li and Yingyu Liang. "Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data". In: *NeurIPS* (2018).

[174] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. "Algorithmic Regularization in Over-parameterized Matrix Sensing and Neural Networks with Quadratic Activations". In: *COLT* (2018).

[175] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. "Learning Over-Parametrized Two-Layer ReLU Neural Networks beyond NTK". In: *COLT* (2020).

[176] Yuanzhi Li and Yang Yuan. "Convergence Analysis of Two-layer Neural Networks with ReLU Activation". In: *NeurIPS* (2017).

[177] Nick Littlestone and Manfred Warmuth. "The weighted majority algorithm". In: *Information and computation* (1994).

[178] Chaoyue Liu and Mikhail Belkin. "Parametrized Accelerated Methods Free of Condition Number". In: *arXiv:1802.10235* (2018).

[179] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. "On the linearity of large non-linear models: when and why the tangent kernel is constant". In: *arXiv:2010.01092* (2020).

[180] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. "Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning". In: *arXiv:2003.00307* (2020).

[181] Yanli Liu, Yuan Gao, and Wotao Yin. "An Improved Analysis of Stochastic Gradient Descent with Momentum". In: *NeurIPS* (2020).

[182] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. "On the computational efficiency of training neural networks". In: *NIPS* (2014).

[183] Nicolas Loizou and Peter Richtárik. "Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods". In: *arXiv:1712.09677* (2017).

[184] Nicolas Loizou and Peter Richtárik. "Accelerated Gossip via Stochastic Heavy Ball Method". In: *Allerton* (2018).

[185] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *ICLR* (2019).

[186] Haihao Lu and Robert M. Freund. "Generalized Stochastic Frank Wolfe Algorithm with Stochastic "Substitute" Gradient for Structured Convex Optimization". In: *Mathematical Programming* (2020).

[187] Haihao Lu, Robert M. Freund, and Yurii Nesterov. "Relatively-Smooth Convex Optimization by First-Order Methods and Applications". In: *SIAM Journal on Optimization* (2018).

[188] Haihao Lu and Kenji Kawaguchi. "Depth creates no bad local minima". In: *arXiv:1702.08580* (2017).

[189] Haipeng Luo. "Lecture 1". In: *Introduction to Online Learning* (2007).

[190] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. "Adaptive Gradient Methods with Dynamic Bound of Learning Rate". In: *ICLR* (2019).

[191] Kaifeng Lyu and Jian Li. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks". In: *ICLR* (2020).

[192] Vien V. Mai and Mikael Johansson. "Convergence of a Stochastic Gradient Method with Momentum for Non-Smooth Non-Convex Optimization". In: *ICML* (2020).

[193] Brendan McMahan. "A Survey of Algorithms and Analysis for Adaptive Online Learning". In: *JMLR* (2017).

[194] Brendan McMahan and Jacob Abernethy. "Minimax Optimal Algorithms for Unconstrained Linear Optimization". In: *NeurIPS* (2013).

[195] Ting Kei Pong Michael P. Friedlander Ives Macedo. "Gauge optimization and duality". In: *SIAM Journal on Optimization* (2014).

[196] Ioannis Mitliagkas. “Accelerated Methods - Polyak’s Momentum (Heavy Ball Method)”. In: *Online Lecture Note* (2019).

[197] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. “Escaping saddle points in constrained optimization”. In: *NeurIPS* (2018).

[198] Marco Molinaro. “Curvature of Feasible Sets in Offline and Online Optimization”. In: *arXiv:2002.03213* (2020).

[199] Edward Moroshko, Suriya Gunasekar, Blake Woodworth, Jason D. Lee, Nathan Srebro, and Daniel Soudry. “Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy”. In: *NeurIPS* (2020).

[200] Katta G Murty and Santosh N Kabadi. “Some np-complete problems in quadratic and nonlinear programming”. In: *Mathematical programming* (1987).

[201] Cameron Musco, Christopher Musco, and Aaron Sidford. “Stability of the Lanczos method for matrix function approximation”. In: *SODA* (2018).

[202] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel J. Hsu, and Anant Sahai. “Classification vs regression in overparameterized regimes: Does the loss function matter?” In: *arXiv:2005.08054* (2020).

[203] Geoffrey Négiar, Gideon Dresdner, Alicia Tsai, Laurent El Ghaoui, Francesco Locatello, Robert M. Freund, and Fabian Pedregosa. “Stochastic Frank-Wolfe for Constrained Finite-Sum Minimization”. In: *ICML* (2020).

[204] Yuri Nesterov. “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$”. In: *Doklady AN USSR* (1983).

[205] Yuri Nesterov. “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” In: *Soviet Mathematics Doklady* 27 (1983), pp. 372–376.

[206] Yuri Nesterov. “On an approach to the construction of optimal methods of minimization of smooth convex functions”. In: *Ekonom. i. Mat. Metody* 24 (1988), pp. 509–517.

[207] Yuri Nesterov. “Introductory lectures on convex optimization: A basic course”. In: *Springer* (2004).

[208] Yuri Nesterov. “Smooth minimization of nonsmooth functions.” In: *Mathematical programming* (2005).

[209] Yurii Nesterov. “Squared functional systems and optimization problems”. In: *High performance optimization, Springer* (2000).

[210] Yurii Nesterov. “Dual extrapolation and its applications to solving variational inequalities and related problems”. In: *mathematical programming* (2007).

[211] Yurii Nesterov. “Introductory lectures on convex optimization: a basic course”. In: *Springer* (2013).

[212] Yurii Nesterov and B.T. Polyak. “Cubic regularization of Newton method and its global performance”. In: *Math. Program., Ser. A 108, 177–205* (2006).

[213] Jiawang Nie. “The hierarchy of local minimums in polynomial optimization.” In: *Mathematical programming* (2015).

[214] Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. “ipiano: Inertial proximal algorithm for nonconvex optimization”. In: *SIAM Journal of Imaging Sciences* (2014).

[215] Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. “Arkadi Nemirovski”. In: *SIAM Journal on Optimization* (2004).

[216] Francesco Orabona. “A Modern Introduction to Online Learning”. In: *arXiv:1912.13213* (2019).

[217] Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet K. Dokania, and Simon Lacoste-Julien. “Minding the Gaps for Block Frank-Wolfe Optimization for Structural SVMs”. In: *ICML* (2016).

[218] Samet Oymak and Mahdi Soltanolkotabi. “Towards moderate overparameterization: global convergence guarantees for training shallow neural networks”. In: *arXiv:1902.04674* (2019).

[219] Abhishek Panigrahi, Abhishek Shetty, and Navin Goyal. “Effect of Activation Functions on the Training of Overparametrized Neural Nets”. In: *ICLR* (2020).

[220] Neal Parikh and Stephen Boyd. “Proximal Algorithms”. In: *Foundations and Trends in Optimization* (2014).

[221] Mert Pilanci and Tolga Ergen. “Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-layer Networks”. In: *ICML* (2020).

[222] E. S. Polovinkin. “Strongly convex analysis”. In: *Sbornik: Mathematics 187(2):259* (1996).

[223] B.T. Polyak. “Some methods of speeding up the convergence of iteration methods”. In: *USSR Computational Mathematics and Mathematical Physics* (1964).

[224] Boris Polyak. “Gradient methods for minimizing functionals”. In: *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* (1963).

[225] Alexander Rakhlin and Karthik Sridharan. “Statistical Learning Theory and Sequential Prediction”. In: (2016).

[226] Noam Razin and Nadav Cohen. “Implicit regularization in deep learning may not be explainable by norms”. In: *NeurIPS2020* (2020).

[227] Benjamin Recht. “Lyapunov analysis and the Heavy Ball Method”. In: *Lecture note* (2018).

[228] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization.” In: *SIAM Review* (2010).

[229] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. “On the Convergence of Adam and Beyond”. In: *ICLR* (2018).

[230] Sashank Reddi, Manzil Zaheer, Suvrit Sra, Barnabas Poczos, Francis Bach, Ruslan Salakhutdinov, and Alex Smola. “A generic approach for escaping saddle points”. In: *AISTATS* (2018).

[231] R. Tyrrell Rockafellar and Roger J.-B. Wets. “Variational Analysis”. In: *Springer* (1998).

[232] Tyrrell Rockafellar. “Convex Analysis”. In: *Princeton University Press* (1996).

[233] Andrew M Saxe, James L McClelland, and Surya Ganguli. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. In: *ICLR* (2014).

[234] Damien Scieur and Fabian Pedregosa. “Universal Average-Case Optimality of Polyak Momentum”. In: *ICML* (2020).

[235] Shai Shalev-Shwartz. “Online Learning: Theory, Algorithms, and Applications”. In: *The Hebrew University of Jerusalem. PhD. thesis.* (2007).

[236] Shai Shalev-Shwartz et al. “Online learning and online convex optimization”. In: *Foundations and Trends® in Machine Learning* 4.2 (2012), pp. 107–194.

[237] Ohad Shamir. “A Variant of Azuma’s Inequality for Martingales with Subgaussian Tails”. In: *arXiv:1110.2392* (2011).

[238] Ohad Shamir. “Exponential convergence time of gradient descent for one-dimensional deep linear neural networks”. In: *COLT* (2019).

[239] Ohad Shamir and Shai Shalev-Shwartz. “Collaborative Filtering with the Trace Norm: Learning, Bounding, and Transducing”. In: *COLT* (2011).

[240] Yoav Shechtman, Yonina C. Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. “Phase retrieval with application to optical imaging: a contemporary overview.” In: *IEEE signal processing magazine* (2015).

[241] Bin Shi, Simon S. Du, Michael I. Jordan, and Weijie J. Su. “Understanding the Acceleration Phenomenon via High-Resolution Differential Equations”. In: *arXiv:1810.08907* (2018).

[242] David Silver, Julian Schrittwieser, Karen Simonyan, and et al. “Mastering the game of Go without human knowledge”. In: *Nature* (2017).

[243] Mahdi Soltanolkotabi. “Learning Relus via Gradient Descent”. In: *NeurIPS* (2017).

[244] Zhao Song and Xin Yang. “Quadratic suffices for over-parametrization via matrix chernoff bound”. In: *arXiv:1906.03593* (2019).

[245] Nathan Srebro and Adi Shraibman. “Rank, Trace-Norm and Max-Norm”. In: *COLT* (2005).

[246] Matthew Staib, Sashank J. Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. “Escaping Saddle Points with Adaptive Gradient Methods”. In: *ICML* (2019).

[247] Lili Su and Pengkun Yang. “On Learning Over-parameterized Neural Networks: A Functional Approximation Perspective”. In: *NeurIPS* (2019).

[248] Weijie Su, Stephen Boyd, and Emmanuel Candes. “A Differential Equation for Modeling Nesterov’s Accelerated Gradient Method: Theory and Insights.” In: *NeurIPS* (2014).

[249] Ju Sun, Qing Qu, and John Wright. “When Are Nonconvex Problems Not Scary?” In: *NIPS Workshop on Non-convex Optimization for Machine Learning: Theory and Practice* (2015).

[250] Ju Sun, Qing Qu, and John Wright. “A Geometrical Analysis of Phase Retrieval”. In: *International Symposium on Information Theory* (2016).

[251] Tao Sun, Penghang Yin, Dongsheng Li, Chun Huang, Lei Guan, and Hao Jiang. “Non-ergodic Convergence Analysis of Heavy-Ball Algorithms”. In: *AAAI* (2019).

[252] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. “On the importance of initialization and momentum in deep learning”. In: *ICML* (2013).

[253] Kiran K. Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. “Projection Efficient Subgradient Method and Optimal Nonsmooth Frank-Wolfe Method”. In: *ICML* (2020).

[254] Yuandong Tian. “An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis”. In: *ICML* (2017).

[255] Ryan Tibshirani. “Convex Optimization (Lecture 13: February 25)”. In: *Online Lecture Note* (2015).

[256] T. Tieleman and G. Hinton. “RmsProp: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural Networks for Machine Learning* (2012).

[257] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. “Stochastic cubic regularization for fast nonconvex optimization”. In: *NeurIPS* (2018).

[258] Joel A. Tropp. “An Introduction to Matrix Concentration Inequalities”. In: *arxiv:1501.01571* (2014).

[259] Paul Tseng. “On accelerated proximal gradient methods for convex-concave optimization”. In: (2008).

[260] Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. “Matrix Exponentiated Gradient Updates for On-line Learning and Bregman Projection”. In: *JMLR* (2005).

[261] Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. “Matrix Exponentiated Gradient Updates for On-line Learning and Bregman Projection”. In: *Journal of Machine Learning Research* 6 (2005), pp. 995–1018.

[262] Lieven Vandenberghe. “Introduction: Gradient methods”. In: *ECE236C Lecture Note* (2019).

[263] Ashish Vaswani, Noam Shazeer, Niki Parmar, and et al. “Attention is All you Need”. In: *NIPS* (2017).

[264] Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. “Conditional gradient methods for stochastically constrained convex minimization”. In: *ICML* (2020).

[265] Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. “Projection-free Distributed Online Convex Optimization with $O(\sqrt{t})$ Communication Complexity”. In: *ICML* (2020).

[266] Jun-Kun Wang and Jacob Abernethy. “Acceleration through Optimistic No-Regret Dynamics”. In: *NeurIPS* (2018).

[267] Jun-Kun Wang, Bhuvesh Kumar, Jacob Abernethy, and Guanghui Lan. “A Fast Parallelizable Projection-Free Algorithm for the Nuclear-Norm-Ball Constraint”. In: (2021).

[268] Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. “Escaping Saddle Points Faster with Stochastic Momentum”. In: *ICLR* (2020).

[269] Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. “A Modular Analysis of Provable Acceleration via Polyak’s momentum: Training a Wide ReLU Network and a Deep Linear Network”. In: *arXiv:2010.01618* (2021).

[270] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. “Parallel and Distributed Block-Coordinate Frank-Wolfe Algorithms”. In: *ICML* (2016).

[271] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. “Parallel and Distributed Block-Coordinate Frank-Wolfe Algorithms”. In: *ICML* (2016).

[272] Manfred K. K. Warmuth and Dima Kuzmin. “Randomized online pca algorithms with regret bounds that are logarithmic in the dimension”. In: *JMLR* (2008).

[273] Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. “Regularization Matters: Generalization and Optimization of Neural Nets v.s. their Induced Kernel”. In: *NeurIPS* (2019).

[274] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. “A variational perspective on accelerated methods in optimization”. In: *Proceedings of the National Academy of Sciences* 113.47 (2016), E7351–E7358.

[275] Ashia C. Wilson, Michael Jordan, and Benjamin Recht. “A Lyapunov analysis of momentum methods in optimization”. In: *JMLR* (2021).

[276] P. Wolf. “Convergence theory in nonlinear programming”. In: *Integer and Nonlinear Programming* (1970).

[277] Lei Wu, Qingcan Wang, and Chao Ma. “Global Convergence of Gradient Descent for Deep Linear Residual Networks”. In: *NeurIPS* (2019).

[278] Shanshan Wu, Alexandros G Dimakis, and Sujay Sanghavi. "Learning distributions generated by one-layer relu networks". In: *NeurIPS* (2019).

[279] Xiaoxia Wu, Simon S Du, and Rachel Ward. "Global convergence of adaptive gradient methods for an over-parameterized neural network". In: *arXiv:1902.07111* (2019).

[280] Lin Xiao. "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization". In: *JMLR* (2010).

[281] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. "Distance metric learning with application to clustering with side-information". In: *Journal of Machine Learning Research (JMLR)* (2003).

[282] Yi Xu, Jing Rong, and Tianbao Yang. "First-order stochastic algorithms for escaping from saddle points in almost linear time". In: *NeurIPS* (2018).

[283] Greg Yang. "Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation". In: *arXiv:1902.04760* (2019).

[284] Tianbao Yang, Qihang Lin, and Zhe Li. "Unified Convergence Analysis of Stochastic Momentum Methods for Convex and Non-convex Optimization". In: *IJCAI* (2018).

[285] Gilad Yehudai and Ohad Shamir. "Learning a Single Neuron with Gradient Methods". In: *COLT* (2020).

[286] Yiming Ying and Peng Li. "Distance metric learning with eigenvalue optimization". In: *Journal of Machine Learning Research (JMLR)* (2012).

[287] Y. Yu, X. Zhang, and D. Schuurmans. "Generalized Conditional Gradient for Structured Estimation". In: *arXiv:1410.4828* (2014).

[288] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. "Global optimality conditions for deep neural networks". In: *ICLR* (2018).

[289] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. "A Conditional Gradient Framework for Composite Convex Minimization with Applications to Semidefinite Programming". In: *ICML* (2018).

[290] Alp Yurtsever, Joel A. Tropp, Olivier Fercoq, Madeleine Udell, and Volkan Cevher. "Scalable Semidefinite Programming". In: *arXiv:1912.02949* (2019).

[291] Guodong Zhang, James Martens, and Roger B Grosse. "Fast convergence of natural gradient descent for over-parameterized neural networks". In: *NeurIPS* (2019).

[292] Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven C. H. Hoi, and Tong Zhang. "Projection-free distributed online learning in networks". In: *ICML* (2017).

[293] Xinhua Zhang, Dale Schuurmans, and Yao-liang Yu. "Accelerated training for matrixnorm regularization: A boosting approach". In: *NeurIPS* (2012).

[294] Wenjie Zheng, Aurelien Bellet, and Patrick Gallinari. "A distributed Frank–Wolfe framework for learning low-rank matrices with the trace norm". In: *Machine Learning, Springer* (2018).

[295] Kai Zhong, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon. "Recovery Guarantees for One-hidden-layer Neural Networks". In: *ICML* (2017).

[296] Yi Zhou and Yingbin Liang. "Critical points of linear neural networks: Analytical forms and landscape". In: *ICLR* (2018).

[297] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. "Stochastic gradient descent optimizes overparameterized deep relu networks". In: *Machine Learning, Springer* (2019).

[298] Difan Zou and Quanquan Gu. "An Improved Analysis of Training Over-parameterized Deep Neural Networks". In: *NeurIPS* (2019).

[299] Difan Zou, Philip M. Long, and Quanquan Gu. "On the global convergence of training deep linear resnets". In: *ICLR* (2020).